LCP5 LCP-5 CPU CLUSTR DIA
CJKL5BQ

COPYRIGHT (c) 1983-85
AH-T545B-MC
FICHE 01 OF 03

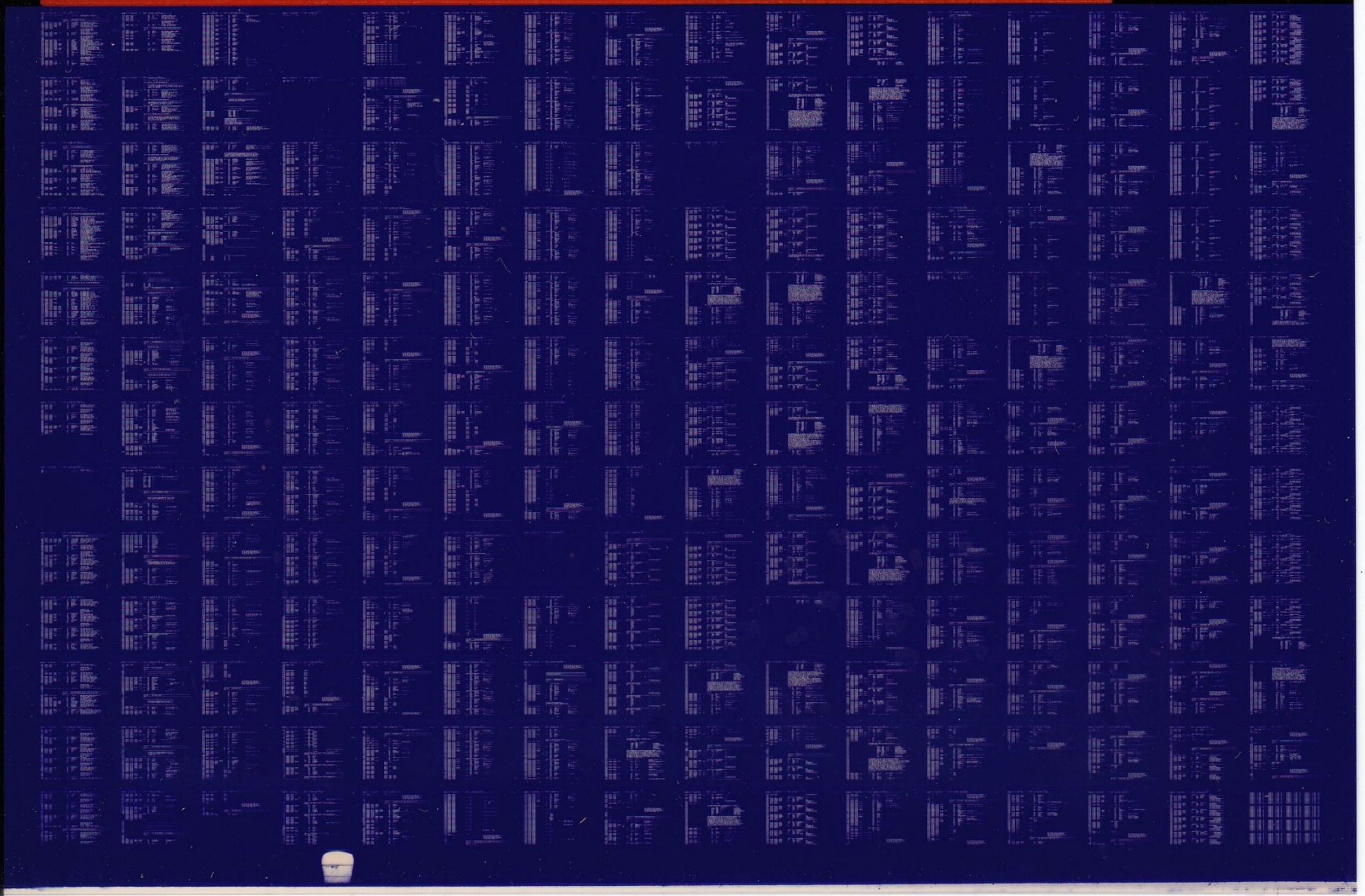
APR 1985
digital
Made In USA

LCP5

LCP-5 CPU CLUSTR DIA
CJKL5BQ

COPYRIGHT (c) 1983-85
AH-T545B-MC
FICHE 02 OF 03

APR 1985
digital
Made In USA



LCP5 LCP-5 CPU CLUSTR DIA
CJKL5B0

COPYRIGHT (c) 1983-85
AH-T545B-MC
FICHE 03 OF 03

APR 1985
digital
Made In USA

```
5708                                .REM_
5709
5710
5711
5712
5713
5714
5715
5716
5717
5718
5719
5720
5721
5722
5723
5724
5725
5726
5727
5728
5729                                          IDENTIFICATION
5730                                          --------------
5731
5732
5733
5734                                PRODUCT CODE:  AC-T544B-MC
5735
5736                                PRODUCT NAME:  CJKL58O LCP-5 CPU CLSTR DIAG
5737
5738                                PRODUCT DATE:  JANUARY 85
5739
5740                                MAINTAINER:  DIAGNOSTIC ENGINEERING
5741
5742
5743
5744
5745                                THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE  WITHOUT  NOTICE
5746                                AND  SHOULD  NOT  BE CONSTRUED AS A COMMITMENT BY DIGITAL CORPORATION.
5747                                DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS
5748                                THAT MAY APPEAR IN THIS DOCUMENT.
5749
5750                                NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON
5751                                EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.
5752
5753                                COPYRIGHT (C):  1983,1985 BY DIGITAL EQUIPMENT CORPORATION
5754
5755                                THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:
5756
5757                                DIGITAL         PDP             UNIBUS          MASSBUS
5758                                DEC             DECUS           DECTAPE         DEC/X11
5759
5760
5761
5762
5763
```

```
                                                    HISTORY
                                                    -------

                        REVISION A                  FIRST RELEASE OF DIAGNOSTIC

                        REVISION B                  THIS REVISION WAS MADE TO CORRECT PROBLEMS
                                                    ENCOUNTERED WHILE RUNNING UNDER APT MODE
                                                    FOR CHANGES, SEE ;DD001
```

TABLE OF CONTENTS

1.0 GENERAL PROGRAM INFORMATION
1.1 ABSTRACT
1.2 SYSTEM REQUIREMENTS
1.3 RELATED DOCUMENTS AND STANDARDS
1.4 DIAGNOSTIC HIERARCHY PREREQUISITES
1.5 ASSUMPTIONS

2.0 OPERATING INSTRUCTIONS
2.1 LOADING AND STARTING PROCEDURE
2.2 PROGRAM OPTIONS
2.3 EXECUTION TIMES

3.0 ERROR INFORMATION
3.1 ERROR REPORTING PROCEDURES
3.2 ERROR HALTS

4.0 PERFORMANCE AND PROGESS REPORTS
4.1 PERFORMANCE REPORTS
4.2 PROGRESS REPORTS

5.0 DEVICE INFORMATION TABLES

6.0 PROGRAM DESCRIPTION
6.1 PROGRAM EXECUTION CHARACTERISTICS
6.2 SUBTEST SUMMARIES
6.3 SPECIAL SUBROUTINE DESCRIPTION

7.0 LISTING

## 1.0 ABSTRACT

THIS PROGRAM IS A GO-NOGO TEST FOR THE MICRO PDP-11 CPU BOARD. IT TESTS THE CPU INCLUDING EIS, THE MMU, THE FPP, THE LTC AND BOTH SLU'S. IT DOES NOT CONTAIN THE CAPABILITIES OF SCOPE LOOPING, ERROR RECOVERY OR PRINTING OF ERROR INFORMATION. ERROR HALTS DO INDICATE WHICH DEVICE FAILED TO ALLOW THE TECHNICIAN TO DETERMINE WHICH DIAGNOSTIC TO USE TO FIX THE BOARD OR WHAT FIELD REPLACEABLE UNIT (FRU) MAY FIX THE BOARD. THE PROGRAM WILL RUN UNDER THE ACT AND APT MANUFACTURING SYSTEMS AND IS CHAINABLE UNDER XXDP+. THIS DIAGNOSTIC WAS CREATED FROM CJKDJB THE KDF11-B CLUSTER DIAGNOSTIC. THE DIFFERENCES ARE THE BDV TESTS ARE FOR 8K ROMS AND UPDATES WERE MADE SO THE PROGRAM WILL RUN UNDER UFD MODE.

## 1.1 SYSTEM REQUIREMENTS

A. HARDWARE REQUIREMENTS

- MICRO PDP-11 CPU MODULE

- 32K OF MEMORY

- THE SECOND SLU MUST HAVE TURN AROUND CONNECTOR.

B. SOFTWARE ENVIRONMENTS

- APT (MULTI-CPU TESTER)

- ACT

- XXDP (SLIDE)

- STAND-ALONE

## 1.2 RELATED DOCUMENTS AND STANDARDS

- ASSEMBLED WITH SYSMAC; SEE FIRST PAGE OF LISTING FOR REVISION NUMBER.

- MXXXX MODULE SPECIFICATION

- DIAGNOSTIC ENGINEERING FUNCTIONAL SPECIFICATION FOR SPECIAL MANUFACTURING TEST BGI-79-003-00-U.

- DIAGNOSTIC ENGINEERING STANDARDS AND CONVENTIONS i75-003-009-02.

## 1.3 PREREQUISITE DIAGNOSTICS

NONE

## 1.4 ASSUMPTIONS

THIS PROGRAM ASSUMES THE MACHINE IS UP SUFFICIENTLY TO ALLOW PROPER OPERATION OF THE MICRO-ODT OF THE DCF11-AA CHIP SET.

THE SYSTEM MUST HAVE PARITY MEMORY LOCATED IN THE FIRST 32K BLOCK.

THE SOFTWARE ASSUMES THAT THERE IS NO MEMORY OR DEVICES LOCATED AT OR BEYOND ADDRESS BIT 17 (64 KW). IF MEMORY IS THERE THE PROGRAM WILL FAIL WHEN IN THE EXTENDED ADDRESS TESTS. IF BIT 7 IS SET IN THE SWITCH REGISTER (176) THIS FAILURE CAN BE PREVENTED SINCE THAT PARTICULAR TEST WILL BE BYPASSED.

SEE PARAGRAPH 2.2.

```
5888
5889
5890                    2.0  OPERATING INSTRUCTIONS
5891
5892
5893
5894
5895                    2.1  LOADING AND STARTING PROCEDURES
5896
5897                         TO LOAD AND START THIS PROGRAM USE THE  STANDARD  PROCEDURES  FOR
5898                         THE DIAGNOSTIC SOFTWARE ENVIRONMENT THAT IS BEING USED.
5899
5900
5901
5902                    2.2  PROGRAM OPTIONS
5903
5904                         THIS PROGRAM USES THE SOFTWARE SWITCH  LOCATION 176 IF PROGRAM IS
5905                         NOT BEING  RUN UNDER  APT MODE (BIT 0 SET OF  LOCATION $ENV).  IF
5906                         PROGRAM IS  BEING RUN IN APT MODE THE LOCATION  $SWREG IN THE APT
5907                         ETABLE IS USED TO STORE OPERATING SWITCHES.
5908
5909
5910                    **********************************************************************************
5911
5912
5913                    WARNING****THIS PROGRAM IS SET TO DO MINIMUM TESTING UNLESS CORRECTIVE
5914                                   ACTION IS TAKEN VIA THE SOFTWARE SWITCH REGISTER (176).
5915                                   BITS 1, 6,7-10 HAVE BEEN SET UP SUCH THAT THE PROGRAM
5916                                   WILL BYPASS CERTAIN TESTS UNLESS THE SWITCH REGISTER
5917                                   BIT IS SET. THIS CONDITION ALSO APPLIES WHEN UNDER CONTROL
5918                                   OF APT. THE APT SWITCH REGISTER, LOCATION 1022, MUST BE
5919                                   CORRECTLY SET AT APT LOAD TIME.
5920
5921
5922                    **********************************************************************************
5923
5924                         BIT #     DEFINITION
5925
5926                         15-11      NOT USED
5927
5928
5929                          10        1 - TEST E102 SWITCHES
5930                                    0 - INHIBIT TESTING E102 SWITCHES
5931
5932                          9         1 - TEST PARITY ERROR DETECTION
5933                                    0 - INHIBIT TESTING PARITY ERROR DETECTION
5934
5935                          8         1 - USE THE Q22BE
5936                                    0 - USE THE QBE IN PLACE OF THE Q22BE
5937
5938                          7         1 - TEST THE UPPER 5 ADDRESS BITS FOR TIME OUT
5939                                    0 - INHIBIT TESTING THE UPPER 5 ADRS BITS
5940
5941                          6         1 - TEST USING A Q BUS EXERCISER (QBE OR Q22BE)
5942                                    0 - INHIBIT TESTS THAT USE A Q BUS EXERCISER
5943
```

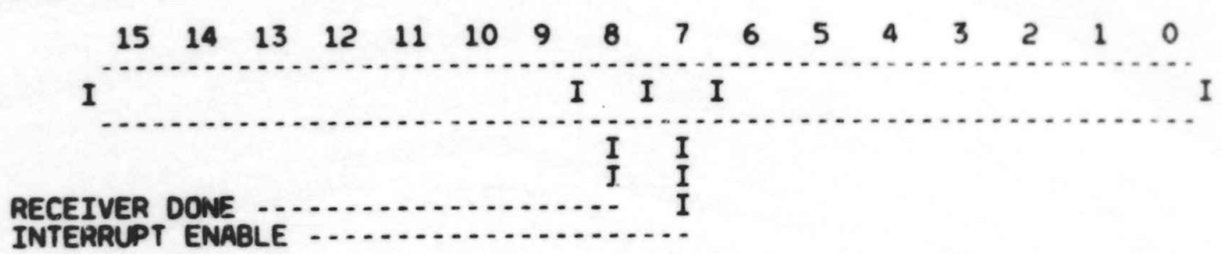
```
                                    5       0 - PROGRAM RESERVED -- PROGRAM WILL CLEAR IF CIS CHIP SET NOT ON BOARD
                                            1 - PROGRAM RESERVED -- PROGRAM WILL SET IF CIS CHIP SET IS ON BOARD
                                    4       0 - TEST SLU2 OF MICRO PDP-11
                                            1 - INHIBIT TESTING OF SLU2

                                    3       0 - TEST LTC OF MICRO PDP-11
                                            1 - INHIBIT TESTING OF LTC

                                    2       0 - TEST SLU1 OF MICRO PDP-11
                                            1 - INHIBIT TESTING OF SLU1

                                    1       1 - TEST FPP INSTRUCTION SET
                                            0 - INHIBIT TESTING OF FPP

                                    0       0 - TEST MEMORY MANAGEMENT UNIT
                                            1 - INHIBIT TESTING OF MEMORY MANAGEMENT UNIT



                          2.3  EXECUTION TIMES

                               FIRST PASS RUNTIME (WORST CASE).....45 SEC
                               LONGEST TEST TIME.....30 SEC
                               ADDITIONAL RUNTIME (EXTRA UNITS).....NONE
                               LONGEST PASS TIME.....45 SEC



                          3.0  ERROR INFORMATION



                          3.1  ERROR REPORTING PROCEDURES

                               THE PROGRAM DOES NOT  TYPE OUT ANY ERROR REPORTS  OF  ITS OWN BUT
                               TAKES ADVANTAGE OF THE  HARDWARE FEATURE THAT TYPES THE PC WHEN A
                               HALT OCCURS.  WHEN AN ERROR IS DETECTED  THE PROGRAM JUMPS TO ONE
                               OF SEVEN HALT  ROUTINES.  THE ROUTINES SIMPLY  MOVE A FATAL ERROR
                               NUMBER INTO LOCATION $FATAL, SET THE FATAL ERROR FLAG IN LOCATION
                               $MSGTY AND EITHER HALT OR IF ON APT DO A BRANCH DOT. THE OPERATOR
                               HAS THREE WAYS  TO DETERMINE THE FAILING  DEVICE; 1) BY EXAMINING
                               LOCATION $FATAL, 2) BY DETERMINING THE  HALT ADDRESS  AND LOOKING
                               UP THE ADDRESS IN THE LISTING AND 3) BY EXAMINING LOCATION $TESTN
                               WHICH WILL CONTAIN THE TEST NUMBER BEING EXECUTED.



                          3.2  ERROR HALTS

                               FOR DISCUSSION SEE SECTION 3.1.  THE LABELS FOR THE HALTS AND THE
                               DEVICE THEY INDICATE HAVING FAILED ARE:

                                    CPUHLT:  CPU
                                    MMUHLT:  MMU
                                    FPPHLT:  FPP
```

```
6000                                        LTCHLT:  LTC
6001                                        SL1HLT:  SLU1
6002                                        SL2HLT:  SLU2
6003                                        EXADHT:  EXT ADRS TEST
6004                                        COMHLT:  SYSTEM INTERACTION
6005                                        Q22HLT:  Q22BE INTERRUPT TEST
6006                                        BDVHLT:  BDV TEST
6007
6008                                    UPON RECEIVING THE ERROR HALT ADDRESS (PC) FROM THE MICRO-ODT THE
6009                                    OPERATOR CAN LOOK UP THESE TAGS IN THE SYMBOL TABLE AT THE END OF
6010                                    THE LISTING TO DETERMINE WHICH HALT WAS EXECUTED.  NOTE:  THE  PC
6011                                    SUPPLIED BY THE MICRO-ODT WILL BE THE HALT ADDRESS PLUS 2.
6012
6013                                    THE OPERATOR CAN DETERMINE WHICH TEST THE ERROR  OCCURRED  IN  BY
6014                                    EXAMINING  LOCATION "$TESTN".  THE TEST NUMBERS EQUATE TO FAILING
6015                                    DEVICES AS FOLLOWS:
6016
6017                                         DEVICE     CONTENTS OF $TESTN
6018
6019                                          CPU            000001
6020                                          MMU            000002
6021                                          FPP            000003
6022                                          SLU1           000004
6023                                          LTC            000005
6024                                          SLU2           000006
6025                                          EXTND ADRS     000007
6026                                          TESTS
6027                                          SYSTEM
6028                                          INTERACTION    000010
6029                                          Q22BE          000011
6030                                          BDV TESTS      000012
6031
6032
6033
6034                                    4.0  PERFORMANCE AND PROGRESS REPORTS
6035
6036                                         THE ONLY REPORT TYPED BY THIS PROGRAM IS  THE  END  PASS  MESSAGE
6037                                         WHICH IS:
6038
6039                                                 CJKL5B0 END OF PASS #XXX
6040
6041                                         WHERE XXX IS THE DECIMAL NUMBER OF PASSES COMPLETED.
6042
6043
6044                                    5.0  DEVICE INFORMATION TABLES
6045
6046                                    SLU1 RCSR
6047
6048                                          15  14  13  12  11  10  9   8   7   6   5   4   3   2   1   0
6049                                         ---------------------------------------------------------------
6050                                         I                               I   I   I                   I
6051                                         ---------------------------------------------------------------
6052                                                                         I   I
6053                                                                         I   I
6054                                     RECEIVER DONE ----------------------    I
6055                                     RECEIVER INTERRUPT ENABLE ---------------
```

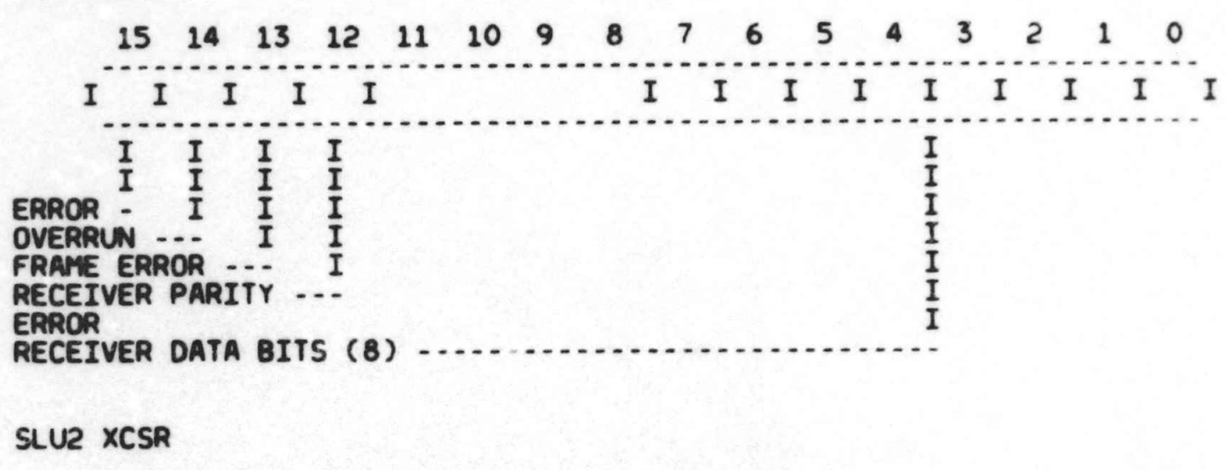
```
6056
6057
6058
6059                    SLU1 RBUF
6060
6061                          15  14  13  12  11  10  9   8   7   6   5   4   3   2   1   0
6062                         ---------------------------------------------------------------
6063                        I   I   I   I   I               I   I   I   I   I   I   I   I   I
6064                         ---------------------------------------------------------------
6065                          I   I   I   I                                   I
6066                          I   I   I   I                                   I
6067                    ERROR-    I   I   I                                   I
6068                    OVERRUN ---   I   I                                   I
6069                    FRAME ERROR ---   I                                   I
6070                    RECEIVE PARITY ----                                   I
6071                    ERROR                                                 I
6072                    RECEIVED DATA BITS (8) ---------------------------------
6073
6074
6075
6076                    SLU1 XCSR
6077
6078                          15  14  13  12  11  10  9   8   7   6   5   4   3   2   1   0
6079                         ---------------------------------------------------------------
6080                      I                         I   I   I                               I
6081                         ---------------------------------------------------------------
6082                                                I   I
6083                    TRANSMITTER READY ----------------  I               I
6084                    TRANSMITTER INTERRUPT ENABLE ----------               I
6085
6086
6087                    SLU1 XBUF
6088
6089                          15  14  13  12  11  10  9   8   7   6   5   4   3   2   1   0
6090                         ---------------------------------------------------------------
6091                        I                           I   I   I   I   I   I   I   I   I
6092                         ---------------------------------------------------------------
6093                                                            I
6094                                                            I
6095                    TRANSMITTER DATA BITS (8) -------------------
6096
6097
6098
6099                    LTC CSR
6100
6101                          15  14  13  12  11  10  9   8   7   6   5   4   3   2   1   0
6102                         ---------------------------------------------------------------
6103                      I                                 I   I
6104                         ---------------------------------------------------------------
6105                                                        I
6106                                                        I
6107                    LINE CLOCK
6108                    INTERRUPT ENABLE ----------------------
6109
6110
6111
```

```
6112
6113
6114
6115
6116
6117
6118
6119
6120
6121
6122
6123
6124
6125
6126
6127
6128
6129
6130
6131
6132
6133
6134
6135
6136
6137
6138
6139
6140
6141
6142
6143
6144
6145
6146
6147
6148
6149
6150
6151
6152
6153
6154
6155
6156
6157
6158
6159
6160
6161
6162
6163
6164
6165
6166
6167
```

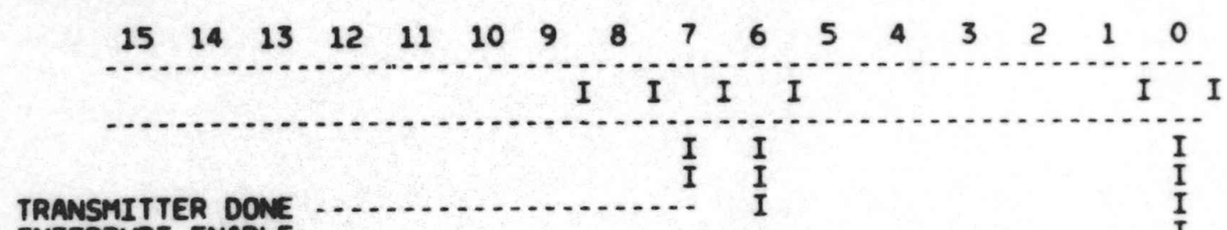
SLU2 RCSR



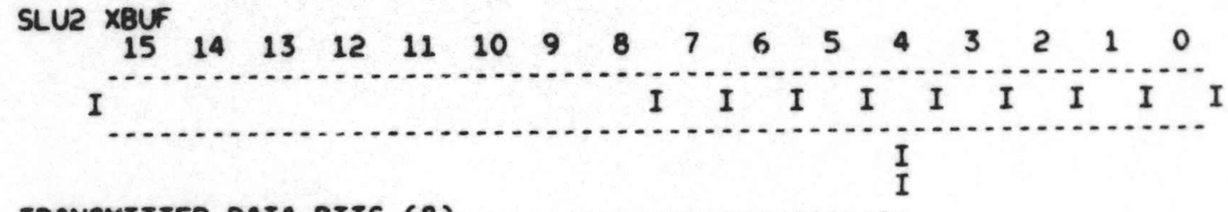

SLU2 RBUF




SLU2 XCSR




SLU2 XBUF




6.0  PROGRAM DESCRIPTION

## 6.1 PROGRAM EXECUTION CHARACTERISTICS

THIS PROGRAM RUNS THE SAME UNDER ALL DIAGNOSTIC MONITORS. WHEN THE TEST IS STARTED AT ADDRESS 200 OCTAL THE TESTING IS DONE AND ON COMPLETION THE TITLE IS TYPED AS PART OF THE END OF PASS MESSAGE.

## 6.2 SUB-TEST SUMMARIES

### 6.2.1 CENTRAL PROCESSING UNIT SUBTEST -

THESE TESTS CHECK THE BASIC INSTRUCTION SET AND ADDRESSING MODES, THE EXTENDED ELEVEN INSTRUCTION SET (EIS) AND TRAPS TESTING. IT IS EQUIVALENT TO CJKDB.

6191
6192
6193
6194
6195
6196
6197
6198
6199
6200
6201
6202
6203
6204
6205
6206
6207
6208
6209
6210
6211
6212
6213
6214
6215
6216
6217
6218
6219
6220
6221
6222
6223
6224
6225
6226
6227
6228
6229
6230
6231
6232
6233
6234
6235
6236
6237
6238
6239
6240
6241
6242
6243
6244
6245
6246

### 6.2.2 MEMORY MANAGEMENT UNIT SUBTEST -

THESE TESTS ARE THE SAME AS IN CJKDA, THE KEF11-AA TEST. THE PROGRAM BEGINS BY TESTING SOME OF THE INTERNAL CPU DATA AND ADDRESS PATHS AND ADDRESS DETECTION LOGIC. NEXT THE MEMORY MANAGEMENT REGISTERS ARE CHECKED FOR DATA RELIABILITY, THEN RELOCATION CAPABILITIES (FORMATION OF A PHYSICAL ADDRESS FROM A VIRTUAL ADDRESS AND ASSOCIATED PAGE DESCRIBTOR (PDR) INFORMATION). FINALLY THE ABORT AND STATUS SEGMENTS OF THE LOGIC ARE CHECKED.

### 6.2.3 EXTENDED ADDRESS BIT TESTING AND PARITY ERROR LOGIC TEST

### 6.2.4 Q22BE BR LEVEL TESTING

### 6.2.5 BDV TESTS

THE BDV TESTS CHECK OUT THE CHECKSUM IN THE DIAGNOSTIC BOOT ROMS OF THE MICRO PDP-11

### 6.2.6 FLOATING POINT PROCESSOR SUBTEST -

THE FLOATING POINT PROCESSOR SUBTEST CHECKS FLOATING POINT REGISTERS FIRST USING A LIMITED NUMBER OF FLOATING POINT INSTRUCTIONS. IT THEN VERIFIES THE REST OF THE FLOATING POINT INSTRUCTION SET USING A NUMBER OF DATA PATTERNS FOR EACH INSTRUCTION.

### 6.2.7 SERIAL LINE UNIT (SLU1) SUBTEST -

THESE TESTS CHECK THE SLU'S REGISTERS FOR ADDRESSING AND DATA HANDLING.

### 6.2.8 LINE TIME CLOCK (LTC) SUBTEST -

FIRST THE REGISTER IS CHECKED FOR ADDRESSING AND BIT SETTING CAPABILITIES THEN THE INTERRUPT LOGIC IS CHECKED. THERE IS ALSO A

### 6.2.9 SERIAL LINE UNIT 2 SUBTEST -

THE TESTING DONE HERE IS SIMILIAR AS FOR THE SLU1. AN EXTERNAL

JUMPER, TURN AROUND CONNECTOR, MUST BE PRESENT.

### 6.2.10 BLAST SUBTEST

THIS TEST CHECKS THE ABILITY OF THE MICRO PDP-11 TO HANDLE SYSTEM INTERACTION. THE CPU HAS TO HANDLE DEVICES AT DIFFERENT PRIORITY LEVELS AND ARBITRATE BETWEEN THEM AND ITS OWN PRIORITY. THE TEST SETS UP ALL DEVICES TO INTERRUPT THEN ENABLES THEM ALL AT ONCE. THE SLU'S TRANSFER DATA UNTIL THEY TRANSFER 400(8) BYTES OR UNTIL ONE SECOND (60 TICKS) OF THE LINE CLOCK HAS BEEN RECEIVED. THE PROGRAM THEN VERIFIES THE NUMBER OF TRANSMITTER INTERRUPTS IS EQUAL TO THE NUMBER OF RECEIVER INTERRUPTS. FINALLY THE DATA TRANSFERRED BY EACH DEVICE IS CHECKED.

## 6.3 SPECIAL SUBROUTINE DESCRIPTIONS

THE ONLY SPECIAL SUBROUTINES ARE THE ERROR ROUTINES EACH SUBTEST HAS IS OWN. THE ROUTINES SIMPLY SET THE FATAL ERROR FLAG IN THE APT MAILBOX AND EITHER "BRANCH SELF" OR "HALT". THIS CHOICE IS DETERMINED IN THE INITIALIZE PORTION OF THE PROGRAM AND IS A "BRANCH SELF" IF RUNNING UNDER APT OR A "HALT" IF RUNNING UNDER ANY OTHER MONITOR.

-

```
                                .TITLE  CJKL58O LCP-5 CPU CLSTR DIAG
                                .ENABLE ABS
                                .NLIST  CND,MC,MD
                                .LIST   ME
            000240              SCOPE=NOP
            000007              R7=%7
            000006              R6=%6
            177776              PS=177776
            177564              TPS=177564
            177566              TPB=177566
            140000              USRM=140000
            030000              PUSRM=30000
            000001              UFDSET=1
            000006              SP=%6
            000006              R6=%6
            000003              TAB=%3
            000001              LAST=%1
            000005              FIRST=%5
            000002              R2=%2
            000000              HLT=HALT
            000003              TRT=3
            000004              ITRAP5=4
            000004              RTRAP5=4            ;RESERVED INST AND ILLEGAL ADDRESSES
            000014              RTRAP4=14           ;FOR TRACE TRAP
            000030              RTRAP3=30           ;FOR EMULATOR TRAP
            000020              RTRAP2=20           ;FOR IOT TRAP
            000034              RTRAP1=34           ;FOR TRAP INST
            177564              TTCSR=177564
            177560              TRCSR=177560
            177562              TKB=177562
            177566              TPB=177566
            000240              BELL=240
            000240              NOP=240
            177776              STATUS=177776
            000077              TRAPA=77
            000010              RTRAP=10
            004700              ILLA=004700
            000100              ILLB=100

            177520              PCR=177520
            177524              LSREG=177524
            177522              RWREG=177522

            177776              CC=177776
            001000              KERSTK=STBOT
            000600              USESTK=STBOT-200
            104377              EMTA=104377
            104777              TRAPC=104777
            000001              APTENV=1
                            .SBTTL  ACT11 HOOKS
 (1)
 (2)                        ;*****************************************************************************************
 (1)                        ;HOOKS REQUIRED BY ACT11
 (1)         000400                 $SVPC=.                 ;SAVE PC
 (1)         000046                 .=46
 (1)  000046 133226                 $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
```

```
 (1)          000052                          .=52
 (1)  000052  000000                          .WORD   0                  ;;2)SET LOC.52 TO ZERO
 (1)          000400                          .=$SVPC                    ;; RESTORE PC
6423          001000                          .=1000
6424                                  .SBTTL  APT MAILBOX-ETABLE
 (1)
 (2)                                  ;*****************************************************************************
 (1)                                  .EVEN
 (1)  001000                          $MAIL:                        ;;APT MAILBOX
 (1)  001000  000000                  $MSGTY: .WORD   AMSGTY        ;;MESSAGE TYPE CODE
 (1)  001002  000000                  $FATAL: .WORD   AFATAL        ;;FATAL ERROR NUMBER
 (1)  001004  000000                  $TESTN: .WORD   ATESTN        ;;TEST NUMBER
 (1)  001006  000000                  $PASS:  .WORD   APASS         ;;PASS COUNT
 (1)  001010  000000                  $DEVCT: .WORD   ADEVCT        ;;DEVICE COUNT
 (1)  001012  000000                  $UNIT:  .WORD   AUNIT         ;;I/O UNIT NUMBER
 (1)  001014  000000                  $MSGAD: .WORD   AMSGAD        ;;MESSAGE ADDRESS
 (1)  001016  000000                  $MSGLG: .WORD   AMSGLG        ;;MESSAGE LENGTH
 (1)  001020                          $ETABLE:                      ;;APT ENVIRONMENT TABLE
 (1)  001020     000                  $ENV:   .BYTE   AENV          ;;ENVIRONMENT BYTE
 (1)  001021     000                  $ENVM:  .BYTE   AENVM         ;;ENVIRONMENT MODE BITS
 (1)  001022  000000                  $SWREG: .WORD   ASWREG        ;;APT SWITCH REGISTER
 (1)  001024  000000                  $USWR:  .WORD   AUSWR         ;;USER SWITCHES
 (1)  001026  000000                  $CPUOP: .WORD   ACPUOP        ;;CPU TYPE,OPTIONS
 (1)                                  ;*                            BITS 15-11=CPU TYPE
 (1)                                  ;*                                   11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
 (1)                                  ;*                                   11/70=06,PDQ=07,Q=10
 (1)                                  ;*                            BIT 10=REAL TIME CLOCK
 (1)                                  ;*                            BIT  9=FLOATING POINT PROCESSOR
 (1)                                  ;*                            BIT  8=MEMORY MANAGEMENT
 (1)  001030                          $ETEND:
 (1)                                  .MEXIT
6425                                  .SBTTL  APT PARAMETER BLOCK
 (1)
 (2)                                  ;*****************************************************************************
 (1)                                  ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
 (2)                                  ;*****************************************************************************
 (1)          001030                          .$X=.   ;;SAVE CURRENT LOCATION
 (1)          000024                          .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
 (1)  000024  000200                          200     ;;FOR APT START UP
 (1)          000044                          .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
 (1)  000044  001030                          $APTHDR ;;POINT TO APT HEADER BLOCK
 (1)          001030                          .=.$X   ;;RESET LOCATION COUNTER
 (2)                                  ;*****************************************************************************
 (1)                                  ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
 (1)                                  ;INTERFACE SPEC.
 (1)
 (1)  001030                          $APTHD:
 (1)  001030  000000                  $HIBTS: .WORD   0             ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
 (1)  001032  001000                  $MBADR: .WORD   $MAIL         ;;ADDRESS OF APT MAILBOX (BITS 0-15)
 (1)  001034  000010                  $TSTM:  .WORD   10            ;;RUN TIM OF LONGEST TEST
 (1)  001036  000025                  $PASTM: .WORD   25            ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
 (1)  001040  000000                  $UNITM: .WORD   0             ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
 (1)  001042  000014                          .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
6426                                  ;*****************************************************************************
6427                                  ;SOME POINTERS TO CPU TRAP HANDLERS
6428                                  ;*****************************************************************************
```

```
        000004                          .=4
000004  021526                          T04
000006  000000                          0
000010  021530                          T010
000012  000000                          0
000014  021532                          T014
000016  000000                          0
000020  021534                          T020
000022  000000                          0
        000034                          .=34
000034  021540                          T034
000036  000000                          0
        000040                          .=40            ;DD001 TRAP CATCHER ADDED
000040  021542                          T040
000042  000000                          0
        000114                          .=114
000114  021544                          T0114
000116  000000                          0
        000244                          .=244
000244  021546                          T0244
000246  000000                          0
000250  021550                          T0250
000252  000000                          0

        000172                  .=172
000172  000000                  MTFLAG:         0       ;MULTI-TESTER ACTIVE BIT
000174  000000                  DISPREG:        0       ;SOFTWARE DISPLAY REGISTER
000176  000024                  SWREG:          24      ;SOFTWARE SWITCH REGISTER

                                ;*******************************************************************************
                                ;DATA TABLE FOR USE IN ADDRESSING MODE TESTS
                                ;*******************************************************************************
        000370                          .=370
000370  000000  000000  000000          0,0,0,0,0,0
000376  000000  000000  000000
000404  000001  000001  177777          1,1,-1
                                ;*******************************************************************************
                                ;SET UP STARTING ADDRESS
        001000                  .=1000
001000  000000                  STBOT:  .WORD 0                         ;STACK POINTER
                                ;* ***************************************************************


        000510                          .=510                           ;Q22BE DEVICE VECTOR
                                                                        ;AREA

        000200                          .=200
000200  000167  000776                  JMP     START
000204  012706  001000                  MOV     #STBOT,R6               ;SET STACK POINTER
000210  012702  001004                  MOV     #$TESTN,R2              ;SET MAILBOX POINTER
000214  000137                          JMP     @(PC)+                  ;JUMP TO SUBTEST
000216  000000                          0                               ;ADDR. OF SUBTEST GOES HERE

        001200                  .=1200
001200  000000                  SAV42:  .WORD   0                       ; LOCATION TO SAVE MONITORS RETURN ADDRESS
                                .SBTTL **STARTING OF CPU TEST **
```

D2



```
 6484  001202                          START:
  (2)                                  ;; LCP/ORION ROUTINE TO SAVE EMTULATOR AND PRIORITY
  (2)
  (2)  001202  005767  000062          EMTSAV: TST     SAV30           ;; FIRST TIME THROUGH ?
  (2)  001206  001034                          BNE     VMKOR           ;; BRANCH IF BEEN HERE ALREADY
  (2)  001210  032737  000040  000052          BIT     #BIT5,@#52      ;; ARE WE IN UFD MODE ?
  (2)  001216  001430                          BEQ     VMKOR           ;; LEAVE IF NOT
  (2)  001220  012767  177777  000046          MOV     #-1,UFDFLG      ;; SET UFD FLAG
  (2)  001226  032737  000100  000052          BIT     #BIT6,@#52      ;; ARE WE IN QUIET MODE ?
  (2)  001234  001403                          BEQ     1$              ;; BR IF NOT
  (2)  001236  012767  177777  000032          MOV     #-1,UQUIET      ;; SET QUIET MODE
  (2)  001244  104042                  1$:     EMT     42              ;; GET ADDRESS OF XXDP DCA TABLE
  (2)  001246  005060  000042                  CLR     42(R0)          ;; CLR XXDP+ "DRSERR"
  (2)  001252  016767  176552  000010          MOV     30,SAV30        ;; SAVE EMULATOR ADDRESS
  (2)  001260  016767  176546  000004          MOV     32,SAV32        ;; SAVE EMULATOR PRIORITY LEVEL
  (2)  001266  000404                          BR      VMKOR           ;; GET AROUND TAG AREA
  (2)  001270  000000                  SAV30:  .WORD   0               ;; PUT EMULATOR INFO HERE
  (2)  001272  000000                  SAV32:  .WORD   0               ;; PUT PRIORITY LOCATION        HERE
  (2)  001274  000000                  UFDFLG: .WORD   0               ;; USER FRIENDLY MODE FLAG
  (2)  001276  000000                  UQUIET: .WORD   0               ;; UFD QUIET MODE FLAG
  (2)  001300                          VMKOR:
  (2)                                  ;*****************************************************************************
  (2)
  (1)
 6485  001300  032737  000040  000052          BIT     #40,@#52                ;DD001
 6486  001306  001403                          BEQ     CLM                     ;DD001
 6487  001310  104042                          EMT     42                      ;DD001
 6488  001312  005060  000042                  CLR     42(R0)                  ;DD001
 6489  001316  012737  000776  000202  CLM:    MOV     #776,@#202              ;DD001
 6490  001324  013767  000042  177646          MOV     @#42,SAV42              ;SAVE MONITOR RETURN ADDRESS
 6491                                          ;MOV    #24,@#176               DD001
 6492  001332  012737  021536  000030  5$:     MOV     #T030,@#30              ;SET UP THE EMT VECTOR
 6493  001340  012737  000000  000032          MOV     #0,@#32                 ;
 6494  001346  032737  000001  001020          BIT     #1,@#$ENV               ;UNDER APT ?
 6495  001354  001010                          BNE     CONTIN                  ;THEN SKIP MESSAGE PRINTOUT ?
 6496  001356  032737  000100  000052          BIT     #100,@#52               ;UNDER UFD ?
 6497  001364  001004                          BNE     CONTIN                  ;THEN SKIP MESSAGE PRINTOUT ?
 6498  001366  012700  133635                  MOV     #STRMSG,R0              ;START MESSAGE
 6499  001372  004767  131766                  JSR     PC,TYPE
 6500  001376  012737  000000  001006  CONTIN: MOV     #0,@#$PASS      ;CLEAR PASS COUNT
 6501  001404  012737  133314  000024  RESTRT: MOV     #PWRDN,@#24             ;SET UP FOR POWER FAIL
 6502  001412  012706  001000                  MOV     #STBOT,R6       ;SET UP STACK
 6503  001416  012737  001446  000004          MOV     #1$,@#4         ;SET UP FOR TIMEOUT IF NO MULTI TESTER
 6504  001424  012737  000340  000006          MOV     #340,@#6
 6505  001432  012737  000002  164000          MOV     #2,@#164000     ;SET BIT1 FOR MULTI TESTER
 6506  001440  012737  000001  000172          MOV     #1,@#MTFLAG     ;SET FLAG TO INDICATE MULTI-TESTER
 6507  001446  012737  021526  000004  1$:     MOV     #T04,@#4                ;SET TRAP CATCHER
 6508  001454  012737  000000  000006          MOV     #0,@#6          ;SET HALT BACK IN LOCATION 6
 6509  001462  012706  001000                  MOV     #STBOT,R6       ;INITIALIZE STACK POINTER
 6510  001466  012737  000001  001004          MOV     #1,@#$TESTN     ;SET TEST NUMBER TO 1
 6511  001474  012737  000000  001002          MOV     #0,@#$FATAL     ;CLEAR ERROR INDICATOR
 6512  001502  012737  000000  001000          MOV     #0,@#$MSGTY     ;CLEAR MESSAGE TYPE(FOR APT)
 6513
 6514  001510  005067  176010                  CLR     LSREG           ;THIS WILL TURN ON THE 4 LEDS
 6515
 6516  001514  105737  001020                  TSTB    @#$ENV          ;RUNNING ON APT
```

```
001520  001036                          BNE     TS1             ;IF YES DO BRANCH SELF ON ERROR
001522  012737  000000  002346          MOV     #0,@#CPUHLT     ;IF NOT THEN PUT A HALT IN ON ERROR
001530  012737  000000  050706          MOV     #0,@#MMUHLT
001536  012737  000000  051626          MOV     #0,@#EXADHT
001544  012737  000000  053442          MOV     #0,@#Q22HLT
001552  012737  000000  124512          MOV     #0,@#FPHLT
001560  012737  000000  126664          MOV     #0,@#LTCHLT
001566  012737  000000  125740          MOV     #0,@#SL1HLT
001574  012737  000000  131534          MOV     #0,@#SL2HLT
001602  012737  000000  132520          MOV     #0,@#COMHLT
001610  012737  000000  054442          MOV     #0,@#BDVHLT

                                ;*******************************************************************************
                                ;
                                ;       THIS TEST EXECUTES EVERY POSSIBLE BRANCH WITH EVERY POSSIBLE
                                ;CONDITION CODE COMBINATION.
                                ;       THE ROUTINE USES TWO TABLES.  THE BRANCH TABLE HOLDS ALL THE
                                ;POSSIBLE BRANCH INSTRUCTIONS, THE OTHER TABLE (YNTAB) HOLDS BIT MAPS FOR
                                ;EACH BRANCH.  A ONE IN THE BIT MAP INDICATES THAT THE CORRESPONDING
                                ;BRANCH INSTRUCTION SHOULD BRANCH FOR THE CONDITION CODE SETTING WHICH
                                ;CORRESPONDS TO THE BIT POSITION WITHIN THE MAP.  FOR EXAMPLE IF THE LEFT
                                ;MOST BIT IS A ONE THEN THE CORRESPONDING BRANCH INSTRUCTION SHOULD BRANCH
                                ;WHEN THE CONDITION CODES ARE 0.
                                ;       THE ROUTINE CONSISTS OF NESTED LOOPS; THE OUTER LOOP SETS UP
                                ;ALL THE POSSIBLE BRANCH INSTRUCTIONS.  THE INNER LOOP SETS UP EVERY POSSIBLE
                                ;CONDITION CODE FOR EACH BRANCH.
                                ;       THE BIT MAP IS USED TO SET THE ADDRESS LOCATION IN TWO
                                ;JUMP MODE 3 INSTRUCTIONS.  THE ADDRESSES ARE CHANGED TO ALLOW THE
                                ;PROGRAM TO CONTINUE OR JUMP TO AN ERROR ROUTINE DEPENDING UPON
                                ;WHETHER IT HANDLED THE BRANCH INSTRUCTION CORRECTLY.
                                ;       AT ANY ERROR HALT, LOCATION, BRH, HOLDS THE BRANCH INSTRUCTION
                                ;UNDER TEST AND LOCATION, CC, HOLDS THE VALUE OF THE CONDITION CODES
                                ;AT THE TIME THE BRANCH WAS EXECUTED.
                                ;
                                ;*******************************************************************************
                                ;TEST 1 TEST THE BRANCH ROM
                                ;*******************************************************************************
001616                          TS1:
001616  012700  021432          SETUP:  MOV     #BRTAB,R0       ;INITIALIZE BRANCH TABLE POINTER
001622  012704  021470                  MOV     #YNTAB,R4       ;INITIALIZE YES/NO BRANCH MAP POINTER
001626  012767  000017  000130          MOV     #15.,BRCT       ;INITIALIZE BRANCH TABLE COUNT
001634  012067  000110          SETBR:  MOV     (R0)+,BRH       ;GET NEXT BRANCH INST.
001640  012401                          MOV     (R4)+,R1        ;GET NEXT BRANCH MAP
001642  012767  177777  000074          MOV     #-1,CC1         ;INITIALIZE CONDITION CODE VALUE
001650  012703  000020                  MOV     #16.,R3         ;INITIALIZE CONDITION CODE COUNT
001654  005267  000064          SETCC:  INC     CC1             ;SET FOR NEXT CC VALUE
001660  032701  100000                  BIT     #100000,R1      ;SEE IF SHOULD BR W/ THESE CC'S
001664  013705  177776                  MOV     @#177776,R5     ;SIMULATE A JNE
001670  042705  177773                  BIC     #177773,R5      ;       (JUMP NOT EQUAL)
001674  000165  001700                  JMP     .+4(R5)         ;       TO SET2BR
001700  000167  000020                  JMP     SET2BR
001704  012767  001766  000042          MOV     #CONT,NBR       ;SET TO CONTINUE IF NO BRANCH
001712  012767  001762  000040          MOV     #ER,YBR         ;SET TO REPORT ERROR IF BRANCH
001720  000167  000014                  JMP     AROUN           ;GO AROUND OPPOSITE CONDITION
001724  012767  001762  000022  SET2BR: MOV     #ER,NBR         ;SET TO REPORT ERROR IF NO BRANCH
001732  012767  001766  000020          MOV     #CONT,YBR       ;SET TO CONTINUE IF BRANCH
```

```
 6567  001740  006101                AROUN:  ROL     R1              ;UPDATE BIT MAP
 6568
 6569  001742  012737                        MOV     (PC)+,@(PC)+    ;SET CONDITION CODE
 6570  001744  000000                CC1:    0                       ;NEW CC VALUE GOES HERE
 6571  001746  177776                        177776
 6572  001750  000000                BRH:    0                       ;BRANCH INST. GOES HERE
 6573  001752  000137                        JMP     @(PC)+          ;THIS JUMP IF NO BRANCH
 6574  001754  000000                NBR:    0                       ;WHERE TO GO IF NO BRANCH OCCURS
 6575  001756  000137                        JMP     @(PC)+          ;THIS JUMP IF BRANCH OCCURS
 6576  001760  000000                YBR:    0                       ;WHERE TO GO IF BRANCH OCCURS
 6577  001762                        ER:
  (2)  001762  000551                        BR      ERROR1          ;
 6578  001764  000000                BRCT:   0
 6579  001766  005303                CONT:   DEC     R3              ;CC'S DONE?
 6580  001770  013705  177776               MOV     @#177776,R5     ;SIMULATE A JNE
  (1)  001774  042705  177773               BIC     #177773,R5      ;       (JUMP NOT EQUAL)
  (1)  002000  000165  002004               JMP     .+4(R5)         ;       TO SETCC
  (1)  002004  000167  177644               JMP     SETCC
 6581  002010  005367  177750               DEC     BRCT            ;BR'S DONE?
 6582  002014  013705  177776               MOV     @#177776,R5     ;SIMULATE A JNE
  (1)  002020  042705  177773               BIC     #177773,R5      ;       (JUMP NOT EQUAL)
  (1)  002024  000165  002030               JMP     .+4(R5)         ;       TO SETBR
  (1)  002030  000167  177600               JMP     SETBR
 6583  002034  012700  000357               MOV     #357,R0         ;IF THIS TEST IS DONE SET UP R0 FOR THE NEXT
 6584                                                                ;SEVEN TESTS. THIS IS SAVING 4 LOCATIONS PER
 6585                                                                ;TEST WHICH I NEED BECAUSE BRANCHES WERE OUT
 6586                                                                ;OF BOUNDS.
 6624
 6625
 6626                                ;*****************************************************************************
  (3)                                ;TEST 2 TEST TRAP OF RESERVED INSTRUCTION
  (4)                                ;*****************************************************************************
  (3)  002040                        TS2:
  (1)  002040  012706  001000               MOV     #STBOT,SP       ;INITIALIZE THE STACK POINTER
  (1)  002044  012767  002066  175736       MOV     #1$,RTRAP               ;SET UP NEW PC IN VECTOR
  (1)  002052  010067  175734               MOV     R0,RTRAP+2              ;SET UP NEW PSW IN VECTOR
  (1)  002056  005067  175714               CLR     STATUS          ;CLEAR PRESENT (OLD) STATUS
  (1)  002062  000077                       TRAPA                   ;DO TRAP
  (1)  002064                        8$:
  (3)  002064  000510                       BR      ERROR1          ;INSTRUCTION FAILED TO TRAP
  (1)  002066  020067  175704        1$:    CMP     R0,STATUS       ;IS NEW STATUS CORRECT
  (1)  002072  001105                       BNE     ERROR1          ;NEW STATUS WRONG
  (1)  002074  020627  000774        2$:    CMP     SP,#STBOT-4     ;DID STACK DECREMENT CORRECTLY
  (1)  002100  001102                       BNE     ERROR1          ;STACK DID NOT DECREMENT CORRECTLY
  (1)  002102  022767  002064  176664 3$:    CMP     #8$,STBOT-4     ;WAS PROPER PC SAVED
  (1)  002110  001076                       BNE     ERROR1          ;PROPER PC WAS NOT SAVED
  (1)  002112  005767  176660        4$:    TST     STBOT-2         ;WAS OLD PSW SAVED
  (1)  002116  001073                       BNE     ERROR1          ;WRONG PSW SAVED
  (1)  002120  062706  000004        5$:    ADD     #4,SP           ;RESET STACK POINTER
  (1)  002124  012767  002146  175656       MOV     #6$,RTRAP               ;SET UP NEW PC IN VECTOR
  (1)  002132  005067  175654               CLR     RTRAP+2         ;SET UP NEW PSW IN VECTOR
  (1)  002136  010037  177776               MOV     R0,@#STATUS     ;SET UP OLD STATUS FOR COMPARISON AFTER TRAP
  (1)  002142  000077                       TRAPA                   ;DO TRAP
  (3)  002144  000460                       BR      ERROR1          ;INSTRUCTION FAILED TO TRAP
  (1)  002146  005767  175624        6$:    TST     STATUS          ;IS NEW PSW CORRECT
  (1)  002152  001055                       BNE     ERROR1          ;NEW PSW WRONG
```

```
(1)  002154  020067  176616          7$:     CMP     R0,STBOT-2      ;WAS OLD STATUS STORED
(1)  002160  001052                          BNE     ERROR1          ;OLD STATUS WRONG
6627                                 ;*****************************************************************************
(3)                                  ;TEST 3 TEST TRAP OF TRAP INSTRUCTION
(4)                                  ;*****************************************************************************
(3)  002162                          TS3:
(1)  002162  012706  001000                  MOV     #STBOT,SP       ;INITIALIZE THE STACK POINTER
(1)  002166  012767  002210  175640          MOV     #1$,RTRAP1              ;SET UP NEW PC IN VECTOR
(1)  002174  010067  175636                  MOV     R0,RTRAP1+2             ;SET UP NEW PSW IN VECTOR
(1)  002200  005067  175572                  CLR     STATUS          ;CLEAR PRESENT (OLD) STATUS
(1)  002204  104400                          TRAP                    ;DO TRAP
(1)  002206                          8$:
(3)  002206  000437                          BR      ERROR1          ;INSTRUCTION FAILED TO TRAP
(1)  002210  020067  175562          1$:     CMP     R0,STATUS       ;IS NEW STATUS CORRECT
(1)  002214  001034                          BNE     ERROR1          ;NEW STATUS WRONG
(1)  002216  020627  000774          2$:     CMP     SP,#STBOT-4     ;DID STACK DECREMENT CORRECTLY
(1)  002222  001031                          BNE     ERROR1          ;STACK DID NOT DECREMENT CORRECTLY
(1)  002224  022767  002206  176542  3$:     CMP     #8$,STBOT-4     ;WAS PROPER PC SAVED
(1)  002232  001025                          BNE     ERROR1          ;PROPER PC WAS NOT SAVED
(1)  002234  005767  176536          4$:     TST     STBOT-2         ;WAS OLD PSW SAVED
(1)  002240  001022                          BNE     ERROR1          ;WRONG PSW SAVED
(1)  002242  062706  000004          5$:     ADD     #4,SP           ;RESET STACK POINTER
(1)  002246  012767  002270  175560          MOV     #6$,RTRAP1              ;SET UP NEW PC IN VECTOR
(1)  002254  005067  175556                  CLR     RTRAP1+2                ;SET UP NEW PSW IN VECTOR
(1)  002260  010037  177776                  MOV     R0,@#STATUS     ;SET UP OLD STATUS FOR COMPARISON AFTER TRAP
(1)  002264  104777                          TRAPC                   ;DO TRAP WITH LOWER BYTE ALL ONES
(3)  002266  000407                          BR      ERROR1          ;INSTRUCTION FAILED TO TRAP
(1)  002270  005767  175502          6$:     TST     STATUS          ;IS NEW PSW CORRECT
(1)  002274  001004                          BNE     ERROR1          ;NEW PSW WRONG
(1)  002276  020067  176474          7$:     CMP     R0,STBOT-2      ;WAS OLD STATUS STORED
(1)  002302  001001                          BNE     ERROR1          ;OLD STATUS WRONG
6628 002304  000436                          BR      CPUHLT+34       ;GET OVER ERROR CALL TO NEXT TEST IF NO ERROR
6629                                 ;*****************************************************************************
6630                                 ;THIS ERROR IS USED FOR THE ENTIRE CPU,TRAPS AND EIS PORTION OF
6631                                 ;THIS TEST
6632                                 ;*****************************************************************************
6633 002306  004767  131116          ERROR1: JSR     PC,ABORT                ;CHECK IF WE ARE UNDER UFD ?
6634 002312  012737  000001  001002          MOV     #1,@#$FATAL             ;SET UP FATAL ERROR NUMBER
6635 002320  012767  000001  176452          MOV     #1,$MSGTY               ;SET FATAL ERROR FLAG
6636 002326  032737  000001  001020          BIT     #1,@#ENV                ;UNDER APT
6637 002334  001004                          BNE     CPUHLT                  ;YES, THEN DO NOT PRINT
6638 002336  012700  002350                  MOV     #CPUMSG,R0
6639 002342  004767  131016                  JSR     PC,TYPE                 ;TYPE MSG
6640 002346  000777                  CPUHLT: BR      .
6641
6642 002350  040506  046111  042105  CPUMSG: .ASCIZ  /FAILED DURING CPU TESTS/<12><15>
     002356  042040  051125  047111
     002364  020107  050103  020125
     002372  042524  052123  005123
     002400  000015
6643                                 .EVEN
6644
6645                                 ;*****************************************************************************
(3)                                  ;TEST 4 TEST TRAP OF IOT INSTRUCTION
(4)                                  ;*****************************************************************************
(3)  002402                          TS4:
```

H2



```
(1)  002402  012706  001000                  MOV     #STBOT,SP        ;INITIALIZE THE STACK POINTER
(1)  002406  012767  002430  175404          MOV     #1$,RTRAP2                ;SET UP NEW PC IN VECTOR
(1)  002414  010067  175402                  MOV     R0,RTRAP2+2               ;SET UP NEW PSW IN VECTOR
(1)  002420  005067  175352                  CLR     STATUS           ;CLEAR PRESENT (OLD) STATUS
(1)  002424  000004                          IOT                      ;DO TRAP
(1)  002426                          8$:
(3)  002426  000727                          BR      ERROR1           ;INSTRUCTION FAILED TO TRAP
(1)  002430  020067  175342          1$:     CMP     R0,STATUS        ;IS NEW STATUS CORRECT
(1)  002434  001324                          BNE     ERROR1           ;NEW STATUS WRONG
(1)  002436  020627  000774          2$:     CMP     SP,#STBOT-4      ;DID STACK DECREMENT CORRECTLY
(1)  002442  001321                          BNE     ERROR1           ;STACK DID NOT DECREMENT CORRECTLY
(1)  002444  022767  002426  176322  3$:     CMP     #8$,STBOT-4      ;WAS PROPER PC SAVED
(1)  002452  001315                          BNE     ERROR1           ;PROPER PC WAS NOT SAVED
(1)  002454  005767  176316          4$:     TST     STBOT-2          ;WAS OLD PSW SAVED
(1)  002460  001312                          BNE     ERROR1           ;WRONG PSW SAVED
(1)  002462  062706  000004          5$:     ADD     #4,SP            ;RESET STACK POINTER
(1)  002466  012767  002510  175324          MOV     #6$,RTRAP2                ;SET UP NEW PC IN VECTOR
(1)  002474  005067  175322                  CLR     RTRAP2+2                  ;SET UP NEW PSW IN VECTOR
(1)  002500  010037  177776                  MOV     R0,@#STATUS      ;SET UP OLD STATUS FOR COMPARISON AFTER TRAP
(1)  002504  000004                          IOT                      ;DO TRAP
(3)  002506  000677                          BR      ERROR1           ;INSTRUCTION FAILED TO TRAP
(1)  002510  005767  175262          6$:     TST     STATUS           ;IS NEW PSW CORRECT
(1)  002514  001274                          BNE     ERROR1           ;NEW PSW WRONG
(1)  002516  020067  176254          7$:     CMP     R0,STBOT-2       ;WAS OLD STATUS STORED
(1)  002522  001271                          BNE     ERROR1           ;OLD STATUS WRONG
6646                                 ;*******************************************************************************
(3)                                  ;TEST 5 TEST TRAP OF EMT INSTRUCTION
(4)                                  ;*******************************************************************************
(3)  002524                          TS5:
(1)  002524  012706  001000                  MOV     #STBOT,SP        ;INITIALIZE THE STACK POINTER
(1)  002530  012767  002552  175272          MOV     #1$,RTRAP3                ;SET UP NEW PC IN VECTOR
(1)  002536  010067  175270                  MOV     R0,RTRAP3+2               ;SET UP NEW PSW IN VECTOR
(1)  002542  005067  175230                  CLR     STATUS           ;CLEAR PRESENT (OLD) STATUS
(1)  002546  104000                          EMT                      ;DO TRAP
(1)  002550                          8$:
(3)  002550  000656                          BR      ERROR1           ;INSTRUCTION FAILED TO TRAP
(1)  002552  020067  175220          1$:     CMP     R0,STATUS        ;IS NEW STATUS CORRECT
(1)  002556  001253                          BNE     ERROR1           ;NEW STATUS WRONG
(1)  002560  020627  000774          2$:     CMP     SP,#STBOT-4      ;DID STACK DECREMENT CORRECTLY
(1)  002564  001250                          BNE     ERROR1           ;STACK DID NOT DECREMENT CORRECTLY
(1)  002566  022767  002550  176200  3$:     CMP     #8$,STBOT-4      ;WAS PROPER PC SAVED
(1)  002574  001244                          BNE     ERROR1           ;PROPER PC WAS NOT SAVED
(1)  002576  005767  176174          4$:     TST     STBOT-2          ;WAS OLD PSW SAVED
(1)  002602  001241                          BNE     ERROR1           ;WRONG PSW SAVED
(1)  002604  062706  000004          5$:     ADD     #4,SP            ;RESET STACK POINTER
(1)  002610  012767  002632  175212          MOV     #6$,RTRAP3                ;SET UP NEW PC IN VECTOR
(1)  002616  005067  175210                  CLR     RTRAP3+2                  ;SET UP NEW PSW IN VECTOR
(1)  002622  010037  177776                  MOV     R0,@#STATUS      ;SET UP OLD STATUS FOR COMPARISON AFTER TRAP
(1)  002626  104377                          EMTA                     ;DO TRAP WITH LOWER BYTE ALL ONES
(3)  002630  000626                          BR      ERROR1           ;INSTRUCTION FAILED TO TRAP
(1)  002632  005767  175140          6$:     TST     STATUS           ;IS NEW PSW CORRECT
(1)  002636  001223                          BNE     ERROR1           ;NEW PSW WRONG
(1)  002640  020067  176132          7$:     CMP     R0,STBOT-2       ;WAS OLD STATUS STORED
(1)  002644  001220                          BNE     ERROR1           ;OLD STATUS WRONG
6647 002646  000401                          BR      ERROR2+2         ;WE MUST GET OVER ERROR CALL AT END OF THIS TEST
6648                                 ;*******************************************************************************
```

```
                                ;THIS ERROR IS NEEDED BECAUSE BRANCHES IN TRAP TESTS BEYOND HERE CAN NOT
                                ;REACH ERROR1.
                                ;*****************************************************************************
002650  000616                 ERROR2: BR      ERROR1


                                ;*****************************************************************************
                                ;TEST 6 TEST TRAP OF TRACE-TRAP INSTRUCTION
                                ;*****************************************************************************
002652                          TS6:
002652  012706  001000                  MOV     #STBOT,SP       ;INITIALIZE THE STACK POINTER
002656  012767  002700  175130          MOV     #1$,RTRAP4              ;SET UP NEW PC IN VECTOR
002664  010067  175126                  MOV     R0,RTRAP4+2             ;SET UP NEW PSW IN VECTOR
002670  005067  175102                  CLR     STATUS          ;CLEAR PRESENT (OLD) STATUS
002674  000003                          TRT                     ;DO TRAP
002676                          8$:
002676  000764                  BR      ERROR2          ;INSTRUCTION FAILED TO TRAP
002700  020067  175072          1$:     CMP     R0,STATUS       ;IS NEW STATUS CORRECT
002704  001361                  BNE     ERROR2          ;NEW STATUS WRONG
002706  020627  000774          2$:     CMP     SP,#STBOT-4     ;DID STACK DECREMENT CORRECTLY
002712  001356                  BNE     ERROR2          ;STACK DID NOT DECREMENT CORRECTLY
002714  022767  002676  176052  3$:     CMP     #8$,STBOT-4     ;WAS PROPER PC SAVED
002722  001352                  BNE     ERROR2          ;PROPER PC WAS NOT SAVED
002724  005767  176046          4$:     TST     STBOT-2         ;WAS OLD PSW SAVED
002730  001347                  BNE     ERROR2          ;WRONG PSW SAVED
002732  062706  000004          5$:     ADD     #4,SP           ;RESET STACK POINTER
002736  012767  002760  175050          MOV     #6$,RTRAP4              ;SET UP NEW PC IN VECTOR
002744  005067  175046                  CLR     RTRAP4+2                ;SET UP NEW PSW IN VECTOR
002750  010037  177776                  MOV     R0,@#STATUS     ;SET UP OLD STATUS FOR COMPARISON AFTER TRAP
002754  000003                          TRT                     ;DO TRAP
002756  000734                  BR      ERROR2          ;INSTRUCTION FAILED TO TRAP
002760  005767  175012          6$:     TST     STATUS          ;IS NEW PSW CORRECT
002764  001331                  BNE     ERROR2          ;NEW PSW WRONG
002766  020067  176004          7$:     CMP     R0,STBOT-2      ;WAS OLD STATUS STORED
002772  001326                  BNE     ERROR2          ;OLD STATUS WRONG
                                ;PDP-11 ILLEGAL AND ADDRESS INSTRUCTION TEST
                                ;ALL INSTRUCTIONS THAT ARE RESERVED
                                ;SHOULD TRAP TO LOCATION 4, AND THE
                                ;PC THAT POINTS TO THE TRAPPING INSTRUCTION
                                ;SHOULD BE PLACED ON THE STACK

                                ;*****************************************************************************
                                ;TEST 7 TEST TRAP OF ILLEGAL INSTRUCTION
                                ;*****************************************************************************
002774                          TS7:
002774  012706  001000                  MOV     #STBOT,SP       ;INITIALIZE THE STACK POINTER
003000  012767  003022  174776          MOV     #1$,RTRAP5              ;SET UP NEW PC IN VECTOR
003006  010067  174774                  MOV     R0,RTRAP5+2             ;SET UP NEW PSW IN VECTOR
003012  005067  174760                  CLR     STATUS          ;CLEAR PRESENT (OLD) STATUS
003016  000100                          JMP     @#0                     ;DO TRAP
003020                          8$:
003020  000713                  BR      ERROR2          ;INSTRUCTION FAILED TO TRAP
003022  020067  174750          1$:     CMP     R0,STATUS       ;IS NEW STATUS CORRECT
003026  001310                  BNE     ERROR2          ;NEW STATUS WRONG
003030  020627  000774          2$:     CMP     SP,#STBOT-4     ;DID STACK DECREMENT CORRECTLY
003034  001305                  BNE     ERROR2          ;STACK DID NOT DECREMENT CORRECTLY
```

J2



```
(1)  003036  022767  003020  175730  3$:     CMP     #8$,STBOT-4     ;WAS PROPER PC SAVED
(1)  003044  001301                          BNE     ERROR2          ;PROPER PC WAS NOT SAVED
(1)  003046  005767  175724          4$:     TST     STBOT-2         ;WAS OLD PSW SAVED
(1)  003052  001276                          BNE     ERROR2          ;WRONG PSW SAVED
(1)  003054  062706  000004          5$:     ADD     #4,SP           ;RESET STACK POINTER
(1)  003060  012767  003102  174716          MOV     #6$,RTRAP5              ;SET UP NEW PC IN VECTOR
(1)  003066  005067  174714                  CLR     RTRAP5+2                ;SET UP NEW PSW IN VECTOR
(1)  003072  010037  177776                  MOV     R0,@#STATUS     ;SET UP OLD STATUS FOR COMPARISON AFTER TRAP
(1)  003076  000100                          JMP     @#0                     ;DO TRAP
(3)  003100  000663                          BR      ERROR2          ;INSTRUCTION FAILED TO TRAP
(1)  003102  005767  174670          6$:     TST     STATUS          ;IS NEW PSW CORRECT
(1)  003106  001260                          BNE     ERROR2          ;NEW PSW WRONG
(1)  003110  020067  175662          7$:     CMP     R0,STBOT-2      ;WAS OLD STATUS STORED
(1)  003114  001255                          BNE     ERROR2          ;OLD STATUS WRONG
6663                                  ;*****************************************************************************
(3)                                   ;TEST 10        TEST TRAP OF ALL ILLEGAL INSTRUCTION
(4)                                   ;*****************************************************************************
(3)  003116                           TS10:
(1)  003116  012706  001000                  MOV     #STBOT,SP       ;INITIALIZE THE STACK POINTER
(1)  003122  012767  003144  174654          MOV     #1$,RTRAP5              ;SET UP NEW PC IN VECTOR
(1)  003130  010067  174652                  MOV     R0,RTRAP5+2             ;SET UP NEW PSW IN VECTOR
(1)  003134  005067  174636                  CLR     STATUS          ;CLEAR PRESENT (OLD) STATUS
(1)  003140  004000                          JSR     @#0,@#0                 ;DO TRAP
(1)  003142                           8$:
(3)  003142  000642                          BR      ERROR2          ;INSTRUCTION FAILED TO TRAP
(1)  003144  020067  174626          1$:     CMP     R0,STATUS       ;IS NEW STATUS CORRECT
(1)  003150  001237                          BNE     ERROR2          ;NEW STATUS WRONG
(1)  003152  020627  000774          2$:     CMP     SP,#STBOT-4     ;DID STACK DECREMENT CORRECTLY
(1)  003156  001234                          BNE     ERROR2          ;STACK DID NOT DECREMENT CORRECTLY
(1)  003160  022767  003142  175606  3$:     CMP     #8$,STBOT-4     ;WAS PROPER PC SAVED
(1)  003166  001230                          BNE     ERROR2          ;PROPER PC WAS NOT SAVED
(1)  003170  005767  175602          4$:     TST     STBOT-2         ;WAS OLD PSW SAVED
(1)  003174  001225                          BNE     ERROR2          ;WRONG PSW SAVED
(1)  003176  062706  000004          5$:     ADD     #4,SP           ;RESET STACK POINTER
(1)  003202  012767  003224  174574          MOV     #6$,RTRAP5              ;SET UP NEW PC IN VECTOR
(1)  003210  005067  174572                  CLR     RTRAP5+2                ;SET UP NEW PSW IN VECTOR
(1)  003214  010037  177776                  MOV     R0,@#STATUS     ;SET UP OLD STATUS FOR COMPARISON AFTER TRAP
(1)  003220  004000                          JSR     @#0,@#0                 ;DO TRAP
(3)  003222  000612                          BR      ERROR2          ;INSTRUCTION FAILED TO TRAP
(1)  003224  005767  174546          6$:     TST     STATUS          ;IS NEW PSW CORRECT
(1)  003230  001207                          BNE     ERROR2          ;NEW PSW WRONG
(1)  003232  020067  175540          7$:     CMP     R0,STBOT-2      ;WAS OLD STATUS STORED
(1)  003236  001204                          BNE     ERROR2          ;OLD STATUS WRONG
6664
6665                                  ;*****************************************************************************
6666                                  .SBTTL  DATA PATH TESTS
6667                                  ;
6668                                  ;       THE DATA PATH TESTS ARE USED TO VERIFY THAT VARIOUS
6669                                  ;DATA PATTERNS CAN BE SUCCESSFULLY MOVED THROUGH THE DATA PATHS
6670                                  ;MOVE AND COMPARE MODE 2,3 INSTRUCTIONS ARE USED TO PASS AND
6671                                  ;TEST VARIOUS DATA PATTERNS IN THE DATA PATHS.
6672                                  ;       THE TEST EXERCISES THE INTERNAL DATA PATHS,  AND THE UNIBUS
6673                                  ;DATA TRANSCIEVERS.
6674                                  ;       IF THESE TESTS FAIL, EXAMINE THE TARGET LOCATION (LOC. 0)
6675                                  ;TO SEE WHICH BITS OF THE DATA PATH ARE FAILING.
6676  003240  012737  002306  000030          MOV     #ERROR1,@#30    ;SET UP VECTOR FOR ERROR CALLS
```

K2



```
 6677  003246  012737  000340  000032          MOV     #340,@#32       ;SET UP NEW PSW
 6678  003254  012737  021526  000004          MOV     #TO4,@#4        ;SET UP FOR UNEXPECTED TRAP TO 4
 6679  003262  012737  021530  000010          MOV     #TO10,@#10      ;SET UP FOR UNEXPECTED TRAP TO 10
 6680  003270  012737  021532  000014          MOV     #TO14,@#14      ;SET UP FOR UNEXPECTED TRAP TO 14
 6681  003276  012737  021540  000034          MOV     #TO34,@#34      ;SET UP FOR UNEXPECTED TRAP TO 34
 6682  003304  012737  021542  000040          MOV     #TO40,@#40      ;SET UP FOR UNEXPECTED TRAP TO 40 DD001
 6683
 6684                                  ;*****************************************************************************
 6685                                  ;TEST 11        TEST OF ZEROES IN THE DATA PATH
  (2)                                  ;*****************************************************************************
  (3)                                  TS11:
  (2)  003312
 6686  003312  012737  000000  000000          MOV     #0,@#0          ;MOVE ZEROES THRU ADDRESS LINES, DATA
 6687                                                                  ;LINES AND INTERNAL PATHS
 6688  003320  005737  000000                  TST     @#0             ;SUCCESSFUL?
 6689  003324  001401                          BEQ     TS12
  (3)  003326  104000                          EMT                     ;DATA INCORRECT
 6690
 6691                                  ;*****************************************************************************
 6692                                  ;TEST 12        TEST OF PATTERN 125252 IN DATA PATH
  (2)                                  ;*****************************************************************************
  (3)                                  TS12:
  (2)  003330
 6692  003330  012737  125252  000000          MOV     #125252,@#0     ;MOVE ALTERNATING ONES AND ZEROES
 6693                                                                  ;THRU DATA PATHS
 6694  003336  022737  125252  000000          CMP     #125252,@#0     ;SUCCESSFUL
 6695  003344  001401                          BEQ     TS13
  (3)  003346  104000                          EMT                     ;DATA INCORRECT
 6696
 6697                                  ;*****************************************************************************
 6698                                  ;TEST 13        TEST OF PATTERN 052525 IN DATA PATH
  (2)                                  ;*****************************************************************************
  (3)                                  TS13:
  (2)  003350
 6698  003350  012737  052525  000000          MOV     #052525,@#0     ;MOVE ALTERNATING ZEROES AND ONES
 6699                                                                  ;THRU DATA PATH
 6700  003356  022737  052525  000000          CMP     #052525,@#0     ;SUCCESSFUL?
 6701  003364  001401                          BEQ     TS14
  (3)  003366  104000                          EMT                     ;DATA INCORRECT
 6702
 6703                                  ;*****************************************************************************
 6704                                  ;TEST 14        TEST OF ALL ONES IN DATA PATH
  (2)                                  ;*****************************************************************************
  (3)                                  TS14:
  (2)  003370
 6704  003370  012737  177777  000000          MOV     #177777,@#0     ;MOVE ONES THRU DATA PATH
 6705  003376  022737  177777  000000          CMP     #177777,@#0     ;SUCCESSFUL
 6706  003404  001401                          BEQ     TS15
  (3)  003406  104000                          EMT                     ;DATA INCORRECT
 6707
 6708                                  ;*****************************************************************************
 6709                                  .SBTTL  B-REGISTER TEST
 6710                                  ;
 6711                                  ;       THE B-REGISTER (LOCATION 0) SHIFTING LOGIC TESTS ARE USED
 6712                                  ;TO TEST THAT THE B-REGISTER CAN HOLD VARIOUS DATA PATTERNS AND THAT
 6713                                  ;THE ASSOCIATED LOGIC SUPPORTS THE SHIFTING FUNCTIONS WITHIN THE
 6714                                  ;B-REGISTER AND C-BIT.
 6715                                  ;A ONE IS SHIFTED THROUGH EVERY BIT IN THE B-REGISTER AND C-BIT IN
 6716                                  ;BOTH DIRECTIONS.
```

```
                                ;       THE B-REGISTER ITSELF IS TESTED IN ITS ABILITY AS A BUFFER AND AS
                                ;A SHIFT REGISTER.  DATA IS ALSO PASSED THROUGH THE DATA PATH AND ALU.
                                ;       IF THESE TESTS FAIL, EXAMINE THE TARGET LOCATION (LOC. 0) TO SEE
                                ;WHICH BITS OF THE B-REGISTER MAY BE FAILING.
                                ;
                                ;*****************************************************************************
                                ;TEST 15        SHIFT BIT 0 TO BIT 1
                                ;*****************************************************************************
                                TS15:
003410  000241                          CLC                     ;CLEAR CARRY BIT
003412  012737  000001  000000          MOV     #1,@#0          ;LOAD A 1
003420  006137  000000                  ROL     @#0             ;SHIFT LEFT
003424  022737  000002  000000          CMP     #2,@#0          ;SUCCESSFUL
003432  001401                          BEQ     TS16
003434  104000                          EMT                     ;BIT 1 NOT SET

                                ;*****************************************************************************
                                ;TEST 16        SHIFT CARRY INTO BIT 0
                                ;*****************************************************************************
                                TS16:
003436  012737  000000  000000          MOV     #0,@#0          ;CLEAR LOCATION
003444  000261                          SEC                     ;SET CARRY
003446  006137  000000                  ROL     @#0             ;ROTATE CARRY BIT TO BIT 0
003452  103001                          BCC     CARRY1
003454  104000                          EMT                     ;CARRY CLEAR
003456  022737  000001  000000  CARRY1: CMP     #1,@#0          ;BIT 0 SET
003464  001401                          BEQ     TS17
003466  104000                          EMT                     ;BIT 0 NOT SET

                                ;*****************************************************************************
                                ;TEST 17        LEFT SHIFT FROM BIT 0 TO C-BIT
                                ;*****************************************************************************
                                TS17:
003470  012737  000001  000000          MOV     #1,@#0          ;SET BIT 0
003476  012700  177757                  MOV     #-21,R0         ;SET BIT COUNTER
003502  000241                          CLC                     ;CLEAR C-BIT
003504  005200                  SHL:    INC     R0              ;INCREMENT BIT COUNTER
003506  001404                          BEQ     SHLE            ;BR TO ERROR HALT IF BIT IS LOST
003510  006137  000000                  ROL     @#0             ;SHIFT LEFT ONE POSITION
003514  103373                          BCC     SHL             ;BRANCH IF C-BIT NOT SET
003516  001401                          BEQ     TS20
003520                          SHLE:
003520  104000                          EMT                     ;LEFT SHIFTING LOGIC FAILED

                                ;*****************************************************************************
                                ;TEST 20        SHIFT BIT 15 TO BIT 14
                                ;*****************************************************************************
                                TS20:
003522  012737  100000  000000          MOV     #100000,@#0     ;SET BIT 15
003530  000241                          CLC                     ;CLEAR CARRY
003532  006037  000000                  ROR     @#0             ;SHIFT BIT 15 TO BIT 14
003536  022737  040000  000000          CMP     #40000,@#0      ;SUCCESSFUL
003544  001401                          BEQ     TS21
003546  104000                          EMT                     ;BIT 14 NOT SET

                                ;*****************************************************************************
```

M2



```
                                ;TEST 21        RIGHT SHIFT FROM BIT 15 TO C-BIT
                                ;*************************************************************************************
                                TS21:
003550  012737  100000  000000          MOV     #100000,@#0     ;SET BIT 15
003556  012700  177757                 MOV     #-21,R0         ;SET BIT COUNTER
003562  000241                        CLC                     ;CLEAR C-BIT
003564  005200                SHR:    INC     R0              ;INCREMENT BIT COUNTER
003566  001404                        BEQ     SHRE            ;BR TO ERROR HALT IF BIT IS LOST
003570  006037  000000                ROR     @#0             ;ROTATE RIGHT ONE POSITION
003574  103373                        BCC     SHR             ;BRANCH IF C-BIT CLEAR
003576  001401                        BEQ     TS22
003600                          SHRE:
003600  104000                        EMT                     ;RIGHT SHIFT LOGIC FAILED

                                ;*************************************************************************************
                                .SBTTL  SCRATCH PAD TESTS
                                ;
                                ;       THE SCRATCH PAD TESTS ARE USED TO VERIFY THAT VARIOUS
                                ;DATA PATTERNS CAN BE SUCCESSFULLY HELD IN THE SCRATCH PAD
                                ;CIRCUITRY.  MOVE AND COMPARE INSTRUCTIONS ARE USED TO TEST THAT
                                ;R0 CAN HOLD VARIOUS DATA PATTERNS.  EACH DATA PATTERN IS
                                ;MOVED AND TESTED IN A SMALL LOOP CONVENIENT FOR SCOPING.  THE
                                ;SUCCESSFUL COMPLETION OF THESE TESTS SHOULD VERIFY THE CIRCUITRY EXTERNAL
                                ;TO THE SCRATCH PAD ITSELF.
                                ;       THE REMAINDER OF THE GENERAL REGISTERS ARE TESTED BY MOVING
                                ;A BIT INTO BIT 0 OF THE REGISTER AND SHIFTING IT LEFT ONE
                                ;BIT AT A TIME INTO THE CARRY BIT.  THE RESULT IS THEN CHECKED TO INSURE THAT
                                ;NO BITS WERE PICKED. THE PROCEDURE IS THEN REPEATED UNDER OPPOSITE
                                ;CONDITIONS. THE GENERAL REGISTER AND THE CARRY BIT ARE SET TO
                                ;ALL ONES. AND A ZERO IS SHIFTED LEFT FROM BIT 0 INTO THE CARRY BIT.
                                ;THE RESULT IS THEN CHECKED TO INSURE THAT NO ZEROES WERE PICKED.
                                ;       AT THIS POINT ALL OF THE GENERAL REGISTERS HAVE BEEN EXERCISED
                                ;AS WELL AS REGISTER 11.
                                ;
                                ;*************************************************************************************
                                ;TEST 22        TEST IF R0 CAN HOLD ALL ZEROES
                                ;*************************************************************************************
                                TS22:
003602
003602  012700  000000                MOV     #0,R0           ;MOVE ZEROES TO R0
003606  005700                        TST     R0              ;SUCCESSFUL?
003610  001401                        BEQ     TS23
003612  104000                        EMT                     ;R0 NOT 0

                                ;*************************************************************************************
                                ;TEST 23        TEST IF R0 CAN HOLD ONES AND ZEROES
                                ;*************************************************************************************
                                TS23:
003614
003614  012700  125252                MOV     #125252,R0      ;MOVE ALTERNATING ONES AND ZEROES TO R0
003620  020027  125252                CMP     R0,#125252      ;SUCCESSFUL?
003624  001401                        BEQ     TS24
003626  104000                        EMT                     ;R0 NOT 125252

                                ;*************************************************************************************
                                ;TEST 24        TEST IF R0 CAN HOLD ZEROES AND ONES
                                ;*************************************************************************************
```

```
  (2)  003630
6796   003630  012700  052525          TS24:
6797   003634  020027  052525                  MOV     #052525,R0      ;MOVE ALTERNATING ZEROES AND ONES TO R0
6798   003640  001401                          CMP     R0,#052525      ;SUCCESSFUL?
  (3)  003642  104000                          BEQ     TS25
6799                                           EMT                     ;R0 NOT 52525
6800                                  ;*****************************************************************************
 (2)                                  ;TEST 25        TEST IF R0 CAN HOLD ALL ONES
 (3)                                  ;*****************************************************************************
 (2)  003644                          TS25:
6801   003644  012700  177777                  MOV     #177777,R0      ;MOVE ALL ONES TO R0
6802   003650  020027  177777                  CMP     R0,#177777      ;SUCCESSFUL?
6803   003654  001401                          BEQ     TS26
  (3)  003656  104000                          EMT                     ;R0 NOT 177777
6804
6805                                  ;*****************************************************************************
 (2)                                  ;TEST 26        TEST IF R1 CAN HOLD A ONE IN ALL BITS
 (3)                                  ;*****************************************************************************
 (2)  003660                          TS26:
6806   003660  012701  000001                  MOV     #1,R1           ;SET BIT 0
6807   003664  012700  177757                  MOV     #-21,R0         ;SET BIT COUNTER
6808   003670  000241                          CLC                     ;CLEAR C-BIT
6809   003672  005200                  REG1:   INC     R0              ;INCREMENT BIT COUNTER
6810   003674  001[illegible]                  BEQ     REG1E           ;BR TO ERROR HALT IF BIT IS LOST
6811   003676  006101                          ROL     R1              ;ROTATE 1 POSITION
6812   003700  103374                          BCC     REG1            ;ALL DONE
6813   003702  001401                          BEQ     TS27
 (2)  003704                          REG1E:
 (3)  003704  104000                          EMT                     ;FAILURE WITH R1
6814
6815                                  ;*****************************************************************************
 (2)                                  ;TEST 27        TEST IF R1 CAN HOLD A ZERO IN ALL BITS
 (3)                                  ;*****************************************************************************
 (2)  003706                          TS27:
6816   003706  012701  177776                  MOV     #-2,R1          ;SET ALL ONES IN R1 EXCEPT FOR BIT 0
6817   003712  012700  177757                  MOV     #-21,R0         ;SET BIT COUNTER
6818   003716  000261                          SEC                     ;SET C-BIT
6819   003720  005200                  REG1A:  INC     R0              ;INCREMENT COUNTER
6820   003722  001405                          BEQ     R1ERR           ;BR TO ERROR HALT IF COUNTER=0
6821   003724  006101                          ROL     R1              ;ROTATE 1 POSITION
6822   003726  103774                          BCS     REG1A           ;CONTINUE UNTIL C-BIT IS CLEAR
6823   003730  022701  177777                  CMP     #-1,R1          ;CHECK DATA IN R1
6824   003734  001401                          BEQ     TS30
 (2)  003736                          R1ERR:
 (3)  003736  104000                          EMT                     ;FAILURE WITH R1
6825                                  ;*****************************************************************************
 (2)                                  ;TEST 30        TEST IF R2 CAN HOLD A ONE IN ALL BITS
 (3)                                  ;*****************************************************************************
 (2)  003740                          TS30:
6826   003740  012702  000001                  MOV     #1,R2           ;SET BIT 0
6827   003744  012700  177757                  MOV     #-21,R0         ;SET BIT COUNTER
6828   003750  000241                          CLC                     ;CLEAR C-BIT
6829   003752  005200                  REG2:   INC     R0              ;INCREMENT BIT COUNTER
6830   003754  001403                          BEQ     REG2E           ;BR TO ERROR HALT IF BIT IS LOST
6831   003756  006102                          ROL     R2              ;ROTATE 1 POSITION
6832   003760  103374                          BCC     REG2            ;ALL DONE
```

```
6833  003762  001401                        BEQ     TS31
 (2)  003764
 (3)  003764  104000                REG2E:
                                            EMT                     ;FAILURE WITH R2
6834
6835                                ;*******************************************************************************
 (2)                                ;TEST 31        TEST IF R2 CAN HOLD A ZERO IN ALL BITS
 (3)                                ;*******************************************************************************
 (2)  003766                        TS31:
6836  003766  012702  177776                MOV     #-2,R2          ;SET ALL ONES IN R2 EXCEPT FOR BIT 0
6837  003772  012700  177757                MOV     #-21,R0         ;SET BIT COUNTER
6838  003776  000261                        SEC                     ;SET C-BIT
6839  004000  005200                REG2B:  INC     R0              ;INCREMENT BIT COUNTER
6840  004002  001405                        BEQ     R2ERR           ;BR TO ERROR HALT IF COUNTER=0
6841  004004  006102                        ROL     R2              ;ROTATE 1 POSITION
6842  004006  103774                        BCS     REG2B           ;CONTINUE UNTIL C-BIT IS CLEAR
6843  004010  022702  177777                CMP     #-1,R2          ;CHECK DATA IN R2
6844  004014  001401                        BEQ     TS32
 (2)  004016                        R2ERR:
 (3)  004016  104000                        EMT                     ;FAILURE WITH R2
6845
6846                                ;*******************************************************************************
 (2)                                ;TEST 32        TEST IF R3 CAN HOLD A ONE IN ALL BITS
 (3)                                ;*******************************************************************************
 (2)  004020                        TS32:
6847  004020  012703  000001                MOV     #1,R3           ;SET BIT 0
6848  004024  012700  177757                MOV     #-21,R0         ;SET BIT COUNTER
6849  004030  000241                        CLC                     ;CLEAR C-BIT
6850  004032  005200                REG3:   INC     R0              ;INCREMENT BIT COUNTER
6851  004034  001403                        BEQ     REG3E           ;BR TO ERROR HALT IF BIT IS LOST
6852  004036  006103                        ROL     R3              ;ROTATE 1 POSITION
6853  004040  103774                        BCC     REG3            ;ALL DONE
6854  004042  001401                        BEQ     TS33
 (2)  004044                        REG3E:
 (3)  004044  104000                        EMT                     ;FAILURE WITH R3
6855
6856                                ;*******************************************************************************
 (2)                                ;TEST 33        TEST IF R3 CAN HOLD A ZERO IN ALL BITS
 (3)                                ;*******************************************************************************
 (2)  004046                        TS33:
6857  004046  012703  177776                MOV     #-2,R3          ;SET ALL ONES IN R3 EXCEPT FOR BIT 0
6858  004052  012700  177757                MOV     #-21,R0         ;SET BIT COUNTER
6859  004056  000261                        SEC                     ;SET C-BIT
6860  004060  005200                REG3A:  INC     R0              ;INCREMENT BIT COUNTER
6861  004062  001405                        BEQ     R3ERR           ;BR TO ERROR HALT IF COUNTER=0
6862  004064  006103                        ROL     R3              ;ROTATE 1 POSITION
6863  004066  103774                        BCS     REG3A           ;CONTINUE UNTIL C-BIT IS CLEAR
6864  004070  022703  177777                CMP     #-1,R3          ;CHECK DATA
6865  004074  001401                        BEQ     TS34
 (2)  004076                        R3ERR:
 (3)  004076  104000                        EMT                     ;FAILURE WITH R3
6866
6867                                ;*******************************************************************************
 (2)                                ;TEST 34        TEST IF R4 CAN HOLD A ONE IN ALL BITS
 (3)                                ;*******************************************************************************
 (2)  004100                        TS34:
6868  004100  012704  000001                MOV     #1,R4           ;SET BIT 0
```

```
 6869  004104  012700  177757          MOV     #-21,R0         ;SET BIT COUNTER
 6870  004110  000241                  CLC                     ;CLEAR C-BIT
 6871  004112  005200          REG4:   INC     R0              ;INCREMENT BIT COUNTER
 6872  004114  001403                  BEQ     REG4E           ;BR TO ERROR HALT IF BIT IS LOST
 6873  004116  006104                  ROL     R4              ;ROTATE 1 POSITION
 6874  004120  103374                  BCC     REG4            ;ALL DONE
 6875  004122  001401                  BEQ     TS35
   (2) 004124                  REG4E:
   (3) 004124  104000                  EMT                     ;FAILURE WITH R4
 6876
 6877                      ;*************************************************************************
   (2)                     ;TEST 35        TEST IF R4 CAN HOLD A ZERO IN ALL BITS
   (3)                     ;*************************************************************************
   (2) 004126              TS35:
 6878  004126  012704  177776          MOV     #-2,R4          ;SET ALL ONES IN R4 EXCEPT FOR BIT 0
 6879  004132  012700  177757          MOV     #-21,R0         ;SET BIT COUNTER
 6880  004136  000261                  SEC                     ;SET C-BIT
 6881  004140  005200          REG4A:  INC     R0              ;INCREMENT BIT COUNTER
 6882  004142  001405                  BEQ     R4ERR           ;BR TO ERROR HALT IF COUNTER=0
 6883  004144  006104                  ROL     R4              ;ROTATE 1 POSITION
 6884  004146  103774                  BCS     REG4A           ;CONTINUE UNTIL C-BIT IS CLEAR
 6885  004150  022704  177777          CMP     #-1,R4          ;CHECK DATA
 6886  004154  001401                  BEQ     TS36
   (2) 004156                  R4ERR:
   (3) 004156  104000                  EMT                     ;FAILURE WITH R4
 6887
 6888
 6889                      ;*************************************************************************
   (2)                     ;TEST 36        TEST IF R5 CAN HOLD A ONE IN ALL BITS
   (3)                     ;*************************************************************************
   (2) 004160              TS36:
 6890  004160  012705  000001          MOV     #1,R5           ;SET BIT 0
 6891  004164  012700  177757          MOV     #-21,R0         ;SET BIT COUNTER
 6892  004170  000241                  CLC                     ;CLEAR C-BIT
 6893  004172  005200          REG5:   INC     R0              ;INCREMENT BIT COUNTER
 6894  004174  001403                  BEQ     REG5E           ;BR TO ERROR HALT IF BIT IS LOST
 6895  004176  006105                  ROL     R5              ;ROTATE 1 POSITION
 6896  004200  103374                  BCC     REG5            ;ALL DONE
 6897  004202  001401                  BEQ     TS37
   (2) 004204                  REG5E:
   (3) 004204  104000                  EMT                     ;FAILURE WITH R5
 6898
 6899                      ;*************************************************************************
   (2)                     ;TEST 37        TEST IF R5 CAN HOLD A ZERO IN ALL BITS
   (3)                     ;*************************************************************************
   (2) 004206              TS37:
 6900  004206  012705  177776          MOV     #-2,R5          ;SET ALL ONES IN R5 EXCEPT FOR BIT 0
 6901  004212  012700  177757          MOV     #-21,R0         ;SET BIT COUNTER
 6902  004216  000261                  SEC                     ;SET C-BIT
 6903  004220  005200          REG5A:  INC     R0              ;INCREMENT BIT COUNTER
 6904  004222  001405                  BEQ     R5ERR           ;BR TO ERROR HALT IF COUNTER=0
 6905  004224  006105                  ROL     R5              ;ROTATE 1 POSITION
 6906  004226  103774                  BCS     REG5A           ;CONTINUE UNTIL C-BIT IS C;EAR
 6907  004230  022705  177777          CMP     #-1,R5          ;CHECK DATA
 6908  004234  001401                  BEQ     TS40
   (2) 004236                  R5ERR:
```

```
 (3)  004236  104000                          EMT                     ;FAILURE WITH R5
6909
6910                                  ;*******************************************************************************
 (2)                                  ;TEST 40        TEST IF R6 CAN HOLD A ONE IN ALL BITS
 (3)                                  ;*******************************************************************************
 (2)  004240                          TS40:
6911  004240  012706  000001                  MOV     #1,R6           ;SET BIT 0
6912  004244  012700  177757                  MOV     #-21,R0         ;SET BIT COUNTER
6913  004250  000241                          CLC                     ;CLEAR C-BIT
6914  004252  005200                  REG6:   INC     R0              ;INCREMENT BIT COUNTER
6915  004254  001403                          BEQ     REG6E           ;BR TO ERROR HALT IF BIT IS LOST
6916  004256  006106                          ROL     R6              ;ROTATE 1 POSITION
6917  004260  103374                          BCC     REG6            ;ALL DONE
6918  004262  001401                          BEQ     TS41
 (2)  004264                          REG6E:
 (3)  004264  104000                          EMT                     ;FAILURE WITH R6
6919
6920                                  ;*******************************************************************************
 (2)                                  ;TEST 41        TEST IF R6 CAN HOLD A ZERO IN ALL BITS
 (3)                                  ;*******************************************************************************
 (2)  004266                          TS41:
6921  004266  012706  177776                  MOV     #-2,R6          ;SET ALL ONES IN R6 EXCEPT FOR BIT 0
6922  004272  012700  177757                  MOV     #-21,R0         ;SET BIT COUNTER
6923  004276  000261                          SEC                     ;SET C-BIT
6924  004300  005200                  REG6A:  INC     R0              ;INCREMENT BIT COUNT
6925  004302  001405                          BEQ     R6ERR           ;BR TO ERROR HALT IF COUNTER=0
6926  004304  006106                          ROL     R6              ;ROTATE 1 POSITION
6927  004306  103774                          BCS     REG6A           ;CONTINUE UNTIL C-BIT IS CLEAR
6928  004310  022706  177777                  CMP     #-1,R6          ;CHECK DATA
6929  004314  001401                          BEQ     TS42
 (2)  004316                          R6ERR:
 (3)  004316  104000                          EMT                     ;FAILURE WITH R6
6930
6931                                  ;*******************************************************************************
6932                                  .SBTTL  PSW TESTS
6933                                  ;
6934                                  ;       THE PSW TESTS ARE USED TO VERIFY THAT VARIOUS DATA
6935                                  ;PATTERNS CAN BE SUCCESSFULLY HELD IN THE PSW AND THAT THE
6936                                  ;PSW ADDRESSING LOGIC IS FUNCTIONING.  MOVE AND COMPARE INSTRUCTIONS
6937                                  ;ARE USED TO TEST THAT THE PSW CAN HOLD VARIOUS DATA PATTERNS.
6938                                  ;EACH DATA PATTERN IS MOVED AND TESTED IN A SMALL LOOP CONVENIENT FOR
6939                                  ;SCOPING.
6940                                  ;       THE PSW REGISTER IS TESTED, THE CC INPUTS ARE TESTED
6941                                  ;LATER IN THE MICROCODE TESTS.  SETTING OF THE T-BIT BY THE
6942                                  ;TEST PATTERNS IS PURPOSELY AVOIDED. TESTING OF THE
6943                                  ;T-BIT TRAP CIRCUITRY IS LEFT FOR THE TRAP TEST.
6944                                  ;
6945                                  ;*******************************************************************************
 (2)                                  ;TEST 42        TEST IF PSW WILL HOLD ZEROES
 (3)                                  ;*******************************************************************************
 (2)  004320                          TS42:
6946  004320  012706  001000                  MOV     #STBOT,R6
6947  004324  012737  000000  177776          MOV     #0,@#PS         ;SET PSW TO ZERO
6948  004332  005737  177776                  TST     @#PS            ;SUCCESSFUL
6949  004336  001401                          BEQ     TS43
 (3)  004340  104000                          EMT                     ;PSW NOT 0
```

```
                                        ;*******************************************************************************
                                        ;TEST 43        TEST IF PSW WILL HOLD ONES AND ZEROES
                                        ;*******************************************************************************
 004342                                 TS43:
 004342  012737  000252  177776                 MOV     #252,@#PS       ;MOVE ALT. ONES AND ZEROES TO PSW
 004350  023727  177776  000252                 CMP     @#PS,#252       ;SUCCESSFUL?
 004356  001401                                 BEQ     TS44
 004360  104000                                 EMT                     ;PSW NOT 252

                                        ;*******************************************************************************
                                        ;TEST 44        TEST IF PSW (EXCEPT T-BIT) WILL HOLD ZEROES AND ONES
                                        ;*******************************************************************************
 004362                                 TS44:
 004362  012737  000105  177776                 MOV     #105,@#PS       ;MOVE ALT. ONES AND ZEROES TO PSW
 004370  023727  177776  000105                 CMP     @#PS,#105       ;SUCCESSFUL?
 004376  001401                                 BEQ     TS45
 004400  104000                                 EMT                     ;PSW NOT 105

                                        ;*******************************************************************************
                                        ;TEST 45        TEST IF PSW (EXCEPT T-BIT) WILL HOLD ALL ONES
                                        ;*******************************************************************************
 004402                                 TS45:
 004402  012737  000357  177776                 MOV     #357,@#PS       ;MOVE ONES TO PSW
 004410  023727  177776  000357                 CMP     @#PS,#357       ;SUCCESSFUL
 004416  001401                                 BEQ     TS46
 004420  104000                                 EMT                     ;PSW NOT 357
                                        ;*******************************************************************************
                                        .SBTTL  MICROCODE TESTS
                                        ;
                                        ;       THE TEST EXERCISES  BRANCHES IN THE MICROCODE BY
                                        ;TESTING AT LEAST ONE INSTRUCTION FROM EVERY CLASS OF INSTRUCTION IN
                                        ;ALL POSSIBLE MODES.  FOR EXAMPLE, TO TEST THE SINGLE OPERAND INSTRUCTIONS,
                                        ;AT LEAST ONE SINGLE OPERAND INSTRUCTION IS VERIFIED IN ALL UNIQUE
                                        ;ADDRESSING MODES.  BYTE MODES ARE ALSO TESTED.  AS EACH NEW
                                        ;MODE IS INTRODUCED THE SAME INSTRUCTION IS TRIED AND TESTED IN
                                        ;A SMALL LOOP CONVENIENT FOR SCOPING.  THE TEST IS SET UP USING
                                        ;ONLY INSTRUCTIONS AND ADDRESSING MODES WHICH HAVE BEEN PREVIOUSLY
                                        ;VERIFIED.
                                        ;       IF THESE TESTS FAIL, CHECK THE RESULTS FOR A CLUE TO THE
                                        ;FAULT.
                                        ;
                                        ;*******************************************************************************



                                        ;*******************************************************************************
                                        ;
                                        ;       THE CLR INSTRUCTION IS USED TO INTRODUCE EACH ADDRESSING
                                        ;MODE WITH THE SINGLE OPERAND INSTRUCTION.  FOLLOWING THE SEQUENCE CHECK,
                                        ;THE CLR INSTRUCTION IS EXECUTED AND A BRANCH TEST IS EXECUTED WHICH
                                        ;CHECKS THAT THE Z-BIT WAS PROPERLY SET.  THIS TEST CAN CHECK IR DECODE
                                        ;AND MICROCODE FOR SOP INSTRUCTIONS WITH MODE 0.  FOLLOWING THIS TEST
                                        ;SEVERAL OTHER SOP INSTRUCTIONS ARE INTRODUCED WITH MODE 0.  THESE
                                        ;INSTRUCTIONS MAINPULATE DATA AND SERVE TO CHECK THE DATA RESULTS
                                        ;OF THE SOP INSTRUCTIONS IN THIS TEST.  THE DATA IN THIS TEST IS
```

```
                                ;OPERATED ON BY EACH INSTRUCTION WITHOUT REINITIALIZING.
                                ;
                                ;*****************************************************************************************
                                ;TEST 46        TEST MODE 0 USING SOP INST.
                                ;*****************************************************************************************
                                TS46:
                                        CLR     R0              ;TRY THE CLEAR INST.
                                        BEQ     SOPOA
                                        EMT                     ;CLR DID NOT SET Z-BIT
                                SOPOA:  INC     R0              ;TRY THE INCREMENT INST.
                                        COM     R0              ;TRY COMPLEMENT
                                        INC     R0
                                        BMI     SOP08
                                        EMT                     ;NEGATE DID NOT SET N-BIT
                                SOP08:  COM     R0              ;TRY COMPLEMENT INST.
                                        BEQ     TS47
                                        EMT                     ;CUMMULATIVE RESULT OF CLR,INC,NEG AND COM INSTS. FAILED

                                ;*****************************************************************************************
                                ;
                                ;       THIS TEST INTRODUCES THE REMAINING SOP INSTRUCIONS AND TESTS
                                ;THEM IN MODE 0.  THE PURPOSE IS TO PROVIDE A BASELINE OF
                                ;INSTRUCTIONS FOR USE IN THE SUBSEQUENT TESTS.  SINCE THE MICROCODE FOR
                                ;THESE INSTRUCTIONS IS IDENTICAL TO THAT ALREADY TESTED, ANY TROUBLE
                                ;SHOOTING EFFORTS SHOULD BE AIMED AT THE ACTUAL IR DECODE AND ALU
                                ;FUNCTIONING.
                                ;
                                ;*****************************************************************************************
                                ;TEST 47        TEST REMAINDER OF SOP INSTS IN MODE 0
                                ;*****************************************************************************************
                                TS47:
                                        CLR     R0              ;INITIALIZE
                                        DEC     R0              ;TRY DECREMENT INST.
                                        BMI     SOPOC
                                        EMT                     ;N-BIT NOT SET ON DEC
                                SOPOC:  SEC                     ;INITIALIZE CARRY
                                        ADC     R0              ;TRY ADD CARRY INST
                                        BNE     SOPOD
                                        SEC                     ;INITIALIZE CARRY
                                        SBC     R0              ;TRY SUBTRACT-CARRY INST
                                        BPL     SOPOD
                                        COM     R0
                                        INC     R0
                                        DEC     R0
                                        BEQ     TS50
                                SOPOD:
                                        EMT                     ; CUMMULATIVE RESULT OF ADC,SBC,COM,INC AND DEC INSTS. F

                                ;*****************************************************************************************
                                ;
                                ;       THIS TEST INTRODUCES THE BYTE CONTROL LOGIC OF THE PROCESSOR.
                                ;THE MODE 0 BYTE MICROCODE IS TESTED.  THE METHOD AND SEQUENCE
                                ;OF TESTING IS THE SAME AS THAT USED IN THE SOP MODE 0 TESTS.
                                ;
                                ;*****************************************************************************************
```

```
  (2)                                   ;TEST 50        TEST MODE 0 EVEN BYTE USING SOP INST
  (3)                                   ;*****************************************************************************
  (2)  004506                           TS50:
7038   004506  105000                           CLRB    R0              ;TRY CLEARING EVEN BYTE OF REGISTER
7039   004510  001401                           BEQ     SOPB0A
  (2)  004512  104000                           EMT                     ;CLRB DID NOT SET Z-BIT
7040   004514  105100                   SOPB0A: COMB    R0              ;TRY SETTING EVEN BYTE OF REGISTER
7041   004516  100002                           BPL     SOPB0B
7042   004520  105200                           INCB    R0              ;TRY INCREMENTING EVEN BYTE OF REGISTER>>
7043   004522  001401                           BEQ     TS51
  (2)  004524                           SOPB0B:
  (3)  004524  104000                           EMT                     ;TEST CUMMULATIVE RESULT OF ABOVE BYTE INST.
7044
7045                                    ;*****************************************************************************
7046                                    ;
7047                                    ;       THIS TEST USES THE CLR INSTRUCTION TO INTRODUCE AND TEST
7048                                    ;SINGLE OPERAND MODE 1 INSTRUCTIONS.  AGAIN, THE CLR INSTRUCTION
7049                                    ;IS USED TO INTRODUCE THE MICROCODE AND TO TEST THAT THE PROPER
7050                                    ;CONDITION CODES ARE SET.  OTHER SOP INSTRUCTIONS ARE USED TO MANIPULATE
7051                                    ;COMMON DATA TO VERIFY THAT THE CORRECT DATA IS PRODUCED.
7052                                    ;
7053                                    ;*****************************************************************************
  (2)                                   ;TEST 51        TEST MODE 1 USING SOP INST.
  (3)                                   ;*****************************************************************************
  (2)  004526                           TS51:
7054   004526  005000                           CLR     R0              ;INITIALIZE R0
7055   004530  005010                           CLR     (R0)            ;TRY CLEAR INST W/MODE 1
7056   004532  001401                           BEQ     SOP1A
  (2)  004534  104000                           EMT                     ;CLR DID NOT SET Z-BIT
7057   004536  005310                   SOP1A:  DEC     (R0)            ;TRY DECREMENT INST W/MODE 1
7058   004540  100003                           BPL     SOP1B
7059   004542  000261                           SEC                     ;INITIALIZE CARRY
7060   004544  005510                           ADC     (R0)            ;TRY ADD-CARRY W/MODE 1
7061   004546  001401                           BEQ     TS52
  (2)  004550                           SOP1B:
  (3)  004550  104000                           EMT                     ;TEST CUMMULATIVE RESULT OF ABOVE INST
7062
7063                                    ;*****************************************************************************
7064                                    ;
7065                                    ;       THIS TEST VERIFIES THE BYTE INSTRUCTION MICROCODE FOR MODE 1
7066                                    ;SINGLE OPERAND INSTRUCTIONS.
7067                                    ;       THIS IS THE FIRST PLACE THE SIGN EXTEND LOGIC IS EXERCISED
7068                                    ;AND VERIFIED.
7069                                    ;
7070                                    ;*****************************************************************************
  (2)                                   ;TEST 52        TEST MODE 1 EVEN BYTE USING SOP INST
  (3)                                   ;*****************************************************************************
  (2)  004552                           TS52:
7071   004552  005000                           CLR     R0              ;INITIALIZE R0
7072   004554  005010                           CLR     (R0)            ;INITIALIZE LOC. 0
7073   004556  005110                           COM     (R0)
7074   004560  105010                           CLRB    (R0)            ;TRY TO CLEAR BYTE 0
7075   004562  001401                           BEQ     SOPB1A
  (2)  004564  104000                           EMT                     ;CLRB DID NOT SET Z-BIT
7076   004566  005210                   SOPB1A: INC     (R0)            ;INCREMENT TO TEST WORD
7077   004570  100005                           BPL     SOPB1B
```

```
 7078  004572  105110                COMB    (R0)            ;COMPLEMENT:  ODD BYTE = 376
 7079  004574  105210                INCB    (R0)            ;INC:  ODD BYTE = 377
 7080  004576  100002                BPL     SOPB1B
 7081  004600  105210                INCB    (R0)            ;INCREMENT ODD BYTE=0
 7082  004602  001401                BEQ     TS53
  (2)  004604                   SOPB1B:
  (3)  004604  104000                EMT                     ;CHECK CUMMULATIVE RESULT OF ABOVE INST
 7083
 7084
 7085                           ;*****************************************************************************
 7086                           ;
 7087                           ;       THIS TEST VERIFIES THAT SINGLE OPERAND BYTE INSTRUCTIONS WILL
 7088                           ;FUNCTION CORRECTLY FOR ODD BYTES.
 7089                           ;       THIS IS THE FIRST TIME THAT ADDRESS LINE 0 HAS BEEN
 7090                           ;EXERCISED.  CHECKS ARE MADE THAT THE PROPER BYTE IS MODIFIED AND
 7091                           ;THE CONDITION CODES ARE CHECKED.  IT IS ALSO VERIFIED THAT THE UNADDRESSED
 7092                           ;BYTE IS NOT ALTERED BY THE INSTRUCTION.
 7093                           ;
 7094                           ;*****************************************************************************
  (2)                           ;TEST 53        TEST MODE 1 ODD BYTE USING SOP INST
  (3)                           ;*****************************************************************************
  (2)  004606                   TS53:
 7095  004606  005000                CLR     R0              ;INITIALIZE R0
 7096  004610  005010                CLR     (R0)            ;INITIALIZE LOC. 0
 7097  004612  005110                COM     (R0)
 7098  004614  005200                INC     R0              ;R0=ODD BYTE
 7099  004616  105010                CLRB    (R0)            ;TRY TO CLEAR BYTE 1
 7100  004620  001401                BEQ     SOPB1C
  (2)  004622  104000                EMT                     ;CLRB DID NOT SET Z-BIT
 7101  004624  005300           SOPB1C: DEC  R0              ;R0=WORD ADDR.
 7102  004626  005210                INC     (R0)            ;INCREMENT TO TEST WORD
 7103  004630  005200                INC     R0              ;R0=ODD BYTE
 7104  004632  105110                COMB    (R0)            ;TRY TO COMPLEMENT BYTE 1
 7105  004634  105210                INCB    (R0)
 7106  004636  100002                BPL     SOPB1D
 7107  004640  105210                INCB    (R0)            ;TRY TO INCREMENT BYTE 1
 7108  004642  001401                BEQ     TS54
  (2)  004644                   SOPB1D:
  (3)  004644  104000                EMT                     ;TEST CUMMULATIVE RESULT OF ABOVE INST.
 7109
 7110                           ;*****************************************************************************
 7111                           ;
 7112                           ;       THIS TEST VERIFIES MODE 2 SINGLE-OPERAND INSTRUCTIONS.  PREVIOUSLY
 7113                           ;TESTED INSTRUCTIONS ARE USED TO SET A POINTER IN R0 TO LOC. 400.
 7114                           ;LOC. 400 IS INITIALIZED TO -1 BEFORE A CLR MODE 2 IS EXECUTED.
 7115                           ;       THEN R0 IS DECREMENTED BY TWO TO AGAIN POINT TO 400 BEFORE EACH
 7116                           ;OF SEVERAL MODE 2 INSTRUCTIONS ARE USED TO VERIFY THE DATA RESULTS OF
 7117                           ;THE TEST.  THIS PROCEDURE ALSO VERIFIES THE PROPER INCREMENTING OF THE
 7118                           ;REGISTER.
 7119                           ;
 7120                           ;*****************************************************************************
  (2)                           ;TEST 54        TEST MODE 2 USING SOP INST.
  (3)                           ;*****************************************************************************
  (2)  004646                   TS54:
 7121  004646  005000                CLR     R0              ;SET R0=400
 7122  004650  105100                COMB    R0
```

```
 7123  004652  005200                    INC     R0
 7124  004654  005010                    CLR     (R0)            ;CLEAR 400
 7125  004656  005110                    COM     (R0)            ;INITIALIZE: 400=-1
 7126  004660  005020                    CLR     (R0)+           ;TRY CLEARING WITH MODE 2
 7127  004662  001401                    BEQ     SOPZA
  (2)  004664  104000                    EMT                     ;CLR INST DID NOT SET Z-BIT
 7128  004666  005300            SOPZA:  DEC     R0              ;RESET R0
 7129  004670  005300                    DEC     R0
 7130  004672  005120                    COM     (R0)+           ;TRY COMPLEMENTING WITH MODE 2
 7131  004674  100004                    BPL     SOP2B
 7132  004676  005300                    DEC     R0              ;RESET R0
 7133  004700  005300                    DEC     R0
 7134  004702  005220                    INC     (R0)+           ;TRY INCREMENTING WITH MODE 2
 7135  004704  001401                    BEQ     TS55
  (2)  004706                    SOP2B:
  (3)  004706  104000                    EMT                     ;CHECK CUMMULATIVE RESULT OF ABOVE INST
 7136
 7137                           ;*****************************************************************************
 7138                           ;
 7139                           ;       THIS TEST VERIFIES MODE 2 SINGLE OPERAND INSTRUCTIONS WHICH
 7140                           ;ADDRESS EVEN BYTES.  R0 IS SET TO 400 AND USED TO INITIALIZE LOCATION
 7141                           ;400 TO -1.  CLRB INSTRUCTION IS THEN EXECUTED ON BYTE 400 WITH
 7142                           ;MODE 2.
 7143                           ;       R0 IS THEN DECREMENTED BEFORE EACH OF SEVERAL MODE 2 INSTRUCTIONS
 7144                           ;WHICH ARE USED TO VERIFY THE DATA RESULTS OF THE TEST.  THIS PROCEDURE ALSO
 7145                           ;VERIFIES THE PROPER INCREMENTING OF THE REGISTER.
 7146                           ;
 7147                           ;*****************************************************************************
  (2)                           ;TEST 55        TEST MODE 2 EVEN BYTE USING SOP INST.
  (3)                           ;*****************************************************************************
  (2)  004710                   TS55:
 7148  004710  005000                    CLR     R0              ;SET R0=400
 7149  004712  105100                    COMB    R0
 7150  004714  005200                    INC     R0
 7151  004716  005010                    CLR     (R0)            ;CLEAR 400
 7152  004720  005110                    COM     (R0)            ;INITIALIZE: 400=-1
 7153  004722  105020                    CLRB    (R0)+           ;TRY TO CLEAT 400 W/MODE 2
 7154  004724  001401                    BEQ     SOPB2A
  (2)  004726  104000                    EMT                     ;CLR DID NOT SET Z-BIT
 7155  004730  005300            SOPB2A: DEC     R0              ;RESULT R0=400
 7156  004732  005210                    INC     (R0)            ;INC 400 TO TEST WORD
 7157  004734  105110                    COMB    (R0)
 7158  004736  105220                    INCB    (R0)+           ;TRY TO INC EVEN BYTE
 7159  004740  100003                    BPL     SOPB2B
 7160  004742  005300                    DEC     R0              ;RESET R0=400
 7161  004744  105220                    INCB    (R0)+           ;TRY INCREMENT OF EVEN BYTE
 7162  004746  001401                    BEQ     .S56
  (2)  004750                    SOPB2B:
  (3)  004750  104000                    EMT                     ;TEST CUMMULATIVE RESULT OF ABOVE INST.
 7163
 7164                           ;*****************************************************************************
 7165                           ;
 7166                           ;       THIS TEST FOLLOWS THE SAME PROCEDURE DESCRIBED IN THE PREVIOUS
 7167                           ;TEST.  HERE, THE BYTE INSTRUCTION IS USED TO ADDRESS AN ODD BYTE.
 7168                           ;
 7169                           ;*****************************************************************************
```

```
  (2)
  (3)
  (2)  004752
 7170  004752  005000          ;TEST 56        TEST MODE 2 ODD BYTE USING SOP INST.
 7171  004754  105100          ;*****************************************************************************
 7172  004756  005200          TS56:
 7173  004760  005010                  CLR     R0              ;SET R0=400
 7174  004762  005110                  COMB    R0
 7175  004764  005200                  INC     R0
 7176  004766  105020                  CLR     (R0)            ;CLEAR LOC 400
 7177  004770  001401                  COM     (R0)            ;INITIALIZE: 400=-1
  (2)  004772  104000                  INC     R0              ;R0=ODD BYTE
 7178  004774  005300                  CLRB    (R0)+           ;TRY TO CLEAR ODD BYTE
 7179  004776  005300                  BEQ     SOPB2C
 7180  005000  005220                  EMT                     ;CLRB DID NOT SET Z-BIT
 7181  005002  005300          SOPB2C: DEC     R0              ;R0=WORD ADDR.
 7182  005004  105110                  DEC     R0
 7183  005006  105220                  INC     (R0)+           ;INCREMENT WORD
 7184  005010  100003                  DEC     R0              ;POINT TO ODD BYTE
 7185  005012  005300                  COMB    (R0)            ;COMPLEMENT ODD BYTE
 7186  005014  105220                  INCB    (R0)+           ;TRY TO INCREMENT ODD BYTE
 7187  005016  001401                  BPL     SOPB2D
  (2)  005020                          DEC     R0              ;RESET R0 TO ODD BYTE
  (3)  005020  104000                  INCB    (R0)+           ;TRY TO INCREMENT ODD BYTE
 7188                                  BEQ     TS57
 7189                          SOPB2D:
 7190                                  EMT                     ;TEST CUMMULATIVE RESULT OF ABOVE INST.
 7191
 7192                          ;*****************************************************************************
 7193                          ;
 7194                          ;       THESE TESTS CHECK THE NEGATE INSTRUCTION IN ALL MODES.  PREVIOUSLY
  (2)                     ;TESTED SINGLE-OPERAND INSTRUCTIONS ARE USED TO TEST THE NEGATE INSTRUCTION.
  (3)                     ;
  (2)  005022             ;*****************************************************************************
 7195  005022  005000      ;TEST 57        TEST MODE 0 USING NEGATE INSTRUCTION
 7196  005024  005200      ;*****************************************************************************
 7197  005026  005400      TS57:
 7198  005030  100003              CLR     R0              ;SET R0=0
 7199  005032  001402              INC     R0              ;    R0=1
 7200  005034  102401              NEG     R0              ;TRY NEGATE MODE 0:  R0=-1
 7201  005036  103401              BPL     NEG00           ;CC=1001?
  (1)  005040                      BEQ     NEG00
  (2)  005040  104000              BVS     NEG00
 7202                              BCS     NEG01
 7203  005042  005200      NEG00:
 7204  005044  001401              EMT                     ;NEGATE DID NOT SET CC'S CORRECTLY
  (2)  005046  104000
 7205                      NEG01:  INC     R0              ;TEST DATA RESULT
 7206  005050  105100              BEQ     NEG02
 7207  005052  105400              EMT                     ;DATA RESULT OF NEGATE INCORRECT
 7208  005054  100403
 7209  005056  001402      NEG02:  COMB    R0              ;R0=377
 7210  005060  102401              NEGB    R0              ;R0=1
 7211  005062  103401              BMI     NEG03           ;CC=0001?
  (1)  005064                      BEQ     NEG03
  (2)  005064  104000              BVS     NEG03
                                   BCS     NEG04
                          NEG03:
                                   EMT                     ;NEGB DID NOT SET CC'S CORRECTLY
```

K3



```
                        NEG04:  DEC     R0              ;TEST DATA RESULT
7212  005066  005300            BEQ     TS60
7213  005070  001401            EMT                     ;DATA RESULT OF NEGB INCORRECT
 (3)  005072  104000    ;**********************************************************************
7214                    ;TEST 60        TEST MODE 1 USING NEGATE INST.
 (2)                    ;**********************************************************************
 (3)
 (2)  005074            TS60:
7215  005074  005000            CLR     R0              ;POINT TO LOC. 0
7216  005076  005010            CLR     (R0)            ;CLEAR LOC. 0
7217  005100  005210            INC     (R0)            ;LOC. 0=1
7218  005102  005410            NEG     (R0)            ;TRY NEG. LOC. 0=-1
7219  005104  100003            BPL     NEG10           ;CC=1001
7220  005106  001402            BEQ     NEG10
7221  005110  102401            BVS     NEG10
7222  005112  103401            BCS     NEG11
 (1)  005114            NEG10:
 (2)  005114  104000            EMT                     ;NEGATE DID NOT SET CC'S CORRECTLY
7223
7224  005116  005237  000000  NEG11:  INC     @#0             ;TEST DATA RESULT
7225  005122  001401            BEQ     NEG12
 (2)  005124  104000            EMT                     ;DATA RESULT OF NEGATE INCORRECT
7226  005126  105110    NEG12:  COMB    (R0)            ;LOC. 0=377
7227  005130  105410            NEGB    (R0)            ;TRY NEGB LOC. 0=1
7228  005132  100403            BMI     NEG13           ;CC=0001?
7229  005134  001402            BEQ     NEG13
7230  005136  102401            BVS     NEG13
7231  005140  103401            BCS     NEG14
 (1)  005142            NEG13:
 (2)  005142  104000            EMT                     ;NEGB DID NOT SET CC'S CORRECTLY
7232  005144  005337  000000  NEG14:  DEC     @#0             ;TEST DATA RESULT
7233  005150  001401            BEQ     TS61
 (3)  005152  104000            EMT                     ;DATA RESULT OF NEGB INCORRECT
7234    ;**********************************************************************
 (2)    ;TEST 61        TEST MODE 2 USING NEGATE INSTRUCTION
 (3)    ;**********************************************************************
 (2)  005154            TS61:
7235  005154  005000            CLR     R0              ;POINT TO LOC. 0
7236  005156  005010            CLR     (R0)            ;CLEAR LOC. 0
7237  005160  005210            INC     (R0)            ;LOC. 0=1
7238  005162  005420            NEG     (R0)+           ;TRY NEG.: LOC. 0=-1
7239  005164  100003            BPL     NEG20           ;CC=1001?
7240  005166  001402            BEQ     NEG20
7241  005170  102401            BVS     NEG20
7242  005172  103401            BCS     NEG21
 (1)  005174            NEG20:
 (2)  005174  104000            EMT                     ;NEGATE DID NOT SET CC'S CORRECTLY
7243  005176  105300    NEG21:  DECB    R0              ;R0=LOC. 0
7244  005200  105300            DECB    R0
7245  005202  105420            NEGB    (R0)+           ;BYTE 0=1   R0=1
7246  005204  105420            NEGB    (R0)+           ;BYTE 1=1   R0=2
7247  005206  105340            DECB    -(R0)           ;R0=1  LOC. 0=01
7248  005210  005300            DEC     R0              ;R0=0
7249  005212  001401            BEQ     NEG22
 (2)  005214  104000            EMT                     ;REGISTER NOT INCREMENTED CORRECTLY
7250  005216  005337  000000  NEG22:  DEC     @#0             ;LOC. 0=0
7251  005222  001401            BEQ     TS62
```

```
 (3)  005224  104000                    EMT                     ;NEG BYTE INSTRUCTIONS FAILED
7252
7253                                ;*****************************************************************************
7254                                ;
7255                                ;       THIS TEST VERIFIES MODE 3 SINGLE OPERAND INSTRUCTIONS.  IT
7256                                ;USES LOCATION 0 AS ITS TARGET DATA.  A TABLE LOCATED AT LOC. 400
7257                                ;THRU 402 IS USED TO SUPPLY THE ADDRESS OF LOCATION 0 TO THE
7258                                ;INSTRUCTIONS UNDER TEST.
7259                                ;       R0 IS SET TO 400, THE START OF THE ADDRESS TABLE, AND A CLR
7260                                ;INSTRUCTION IS EXECUTED WITH MODE 3 TO CLEAR LOC. 0.  THEN R0
7261                                ;IS DECREMENTED BY TWO AND TWO OTHER MODE 3 INSTRUCTIONS OPERATE ON
7262                                ;LOC. 0 TO VERIFY THE DATA RESULTS OF THE TEST.  THE PROPER INCREMENTING
7263                                ;OF THE REGISTER IS ALSO VERIFIED IN THIS MANNER.
7264                                ;       IF A FAILURE IS DETECTED BE SURE TO VERIFY THAT THE TABLE
7265                                ;(LOC. 400-402) HAS THE PROPER VALUES (0).
7266                                ;
7267                                ;*****************************************************************************
 (2)                                ;TEST 62        TEST MODE 3 USING SOP INST.
 (3)                                ;*****************************************************************************
 (2)  005226                        TS62:
7268  005226  005000                       CLR     R0              ;SET R0=400
7269  005230  105100                       COMB    R0
7270  005232  005200                       INC     R0
7271  005234  005010                       CLR     (R0)            ;CLEAR LOC 400
7272  005236  005030                       CLR     @(R0)+          ;TRY TO CLEAR LOC 0 USING MODE 3 ;R0=402
7273  005240  001401                       BEQ     SOP3A
 (2)  005242  104000                       EMT                     ;CLR DID NOT SET Z-BIT
7274  005244  005300                SOP3A: DEC     R0              ;RESET R0=400
7275  005246  005300                       DEC     R0
7276  005250  005130                       COM     @(R0)+          ;TRY TO COMPLEMENT LOC 0 OF MODE 3  ;R0=402
7277  005252  100002                       BPL     SOP3B
7278  005254  005230                       INC     @(R0)+          ;TRY TO INCREMENT LOC 0 W/MODE 3  ;R0=404
7279  005256  001401                       BEQ     TS63
 (2)  005260                        SOP3B:
 (3)  005260  104000                       EMT                     ;CUMMULATIVE RESULT OF ABOVE INST FAILED
7280
7281                                ;*****************************************************************************
7282                                ;
7283                                ;       THIS TEST VERIFIES MODE 3 SINGLE OPERAND BYTE INSTRUCTIONS
7284                                ;WHICH ADDRESS EVEN BYTES.  AGAIN, THE TARGET LOCATION 0 IS USED
7285                                ;AND THE SAME TABLE AT 400 IS EMPLOYED.
7286                                ;       AFTER POINTING R4 TO THE TABLE (400) AND SETTING LOCATION
7287                                ;0 TO -1, A CLRB INSTRUCTION IS USED TO CLEAR BYTE 0.
7288                                ;       SEVERAL OTHER MODE 3 INSTRUCTIONS ARE THEN USED WITH THE TABLE
7289                                ;TO VERIFY THE DATA RESULTS AND THE PROPER INCREMENTING OF THE REGISTER.
7290                                ;IF A FAILURE IS DETECTED, BE SURE THAT THE TABLE (LOCATION 400-402) HAS
7291                                ;THE PROPER VALUES (0).
7292                                ;
7293                                ;*****************************************************************************
 (2)                                ;TEST 63        TEST MODE 3 EVEN BYTE USING SOP INST.
 (3)                                ;*****************************************************************************
 (2)  005262                        TS63:
7294  005262  005004                       CLR     R4              ;SET R4=400
7295  005264  105104                       COMB    R4
7296  005266  005204                       INC     R4
7297  005270  005000                       CLR     R0              ;INITIALIZE LOC. 0=-1
```

M3



```
                        T63     TEST MODE 3 EVEN BYTE USING SOP INST.                                   SEQ 0038

 7298  005272  005010                CLR     (R0)
 7299  005274  005110                COM     (R0)            ;LOC. 0=-1
 7300  005276  105034                CLRB    @(R4)+          ;TRY TO CLEAR EVEN BYTE ;LOC. 0=177400  R4=402
 7301  005300  001401                BEQ     SOPB3A
  (2)  005302  104000                EMT                     ;CLRB DID NOT SET Z-BIT
 7302  005304  005304        SOPB3A: DEC     R4              ;RESET POINTER  R4=400
 7303  005306  005304                DEC     R4
 7304  005310  005234                INC     @(R4)+          ;TRY INCREMENTING WORD  LOC.0=177401  R4=402
 7305  005312  100006                BPL     SOPB3B
 7306  005314  105434                NEGB    @(R4)+          ;TRY TO NEGATE EVEN BYTE ;LOC.0=-1 R4=404
 7307  005316  100004                BPL     SOPB3B
 7308  005320  005304                DEC     R4              ;R4=402
 7309  005322  005304                DEC     R4
 7310  005324  105234                INCB    @(R4)+          ;TRY TO INCREMENT EVEN BYTE ;LOC. 0=17400
 7311  005326  001401                BEQ     TS64
  (2)  005330            SOPB3B:
  (3)  005330  104000                EMT                     ;CUMMULATIVE RESULT OF ABOVE INST FAILED
 7312
 7313                   ;*************************************************************************************
 7314                   ;
 7315                   ;       THIS TEST VERIFIES MODE 3 SINGLE OPERAND BYTE INSTRUCTIONS
 7316                   ;WHICH ADDRESS ODD BYTES.  THE TARGET IS BYTE 1.  A TABLE AT
 7317                   ;LOC. 400-406 IS USED.  R0 SERVES AS THE TABLE POINTER.
 7318                   ;       R0 IS INITIALIZED TO 400.  LOC. 0 IS SET TO -1 USING THE
 7319                   ;FIRST TWO TABLE ENTRIES.  A CLRB MODE 3 IS EXECUTED ON BYTE 1 USING
 7320                   ;TABLE ADDRESS AT 404.  R0 IS DECREMENTED TO 402 AND SEVERAL SOP
 7321                   ;MODE 3 INSTRUCTIONS ARE USED TO VERIFY DATA RESULTS AND PROPER
 7322                   ;REGISTER INCREMENTING.
 7323                   ;       THE TABLE (400-406) SHOULD CONTAIN 0,0,1,1 BEFORE AND
 7324                   ;AFTER THE TEST IS RUN.
 7325                   ;
```

N3



```
                                ;*******************************************************************************
                                ;TEST 64        TEST MODE 3 ODD BYTE USING SOP INST.
                                ;*******************************************************************************
 005332                         TS64:
 005332  005000                         CLR     R0              ;SET R0=400
 005334  105100                         COMB    R0
 005336  005200                         INC     R0
 005340  005030                         CLR     @(R0)+          ;INITIALIZE
 005342  005130                         COM     @(R0)+          ;LOC 0=-1 R0=404
 005344  105030                         CLRB    @(R0)+          ;TRY TO CLEAR ODD BYTE LOC. 0=377 R0=406
 005346  001401                         BEQ     SOPB3C
 005350  104000                         EMT                     ;CLRB DID NOT SET Z-BIT
 005352  005300                 SOPB3C: DEC     R0              ;RESET R0=402
 005354  005300                         DEC     R0
 005356  005300                         DEC     R0              ;POINT TO EVEN BYTE ADDR.
 005360  005300                         DEC     R0
 005362  005230                         INC     @(R0)+          ;INCREMENT WORD LOC. 0=400 R0=404
```

```
                                        NEGB    @(R0)+          ;TRY TO NEGATE ODD BYTE LOC. 0-177400 R0-406
                                        BPL     SOPB3D
                                        INCB    @(R0)+          ;TRY TO INCREMENT ODD BYTE LOC.0-0 R0-410
                                        BEQ     TS65
                                SOPB3D:
                                        EMT                     ;CUMMULATIVE RESULT OF ABOVE INSTS FAILED
                                ;*****************************************************************************
                                ;TEST 65        TEST MODE 3 USING NEGATE INSTRUCTION
                                ;*****************************************************************************
                                TS65:
                                        CLR     R0              ;R0-400
                                        COMB    R0
                                        INC     R0
                                        CLR     (R0)            ;LOC. 400-0
                                        CLR     R4              ;R4-0
                                        CLR     (R4)            ;LOC. 0-0
                                        INC     (R4)            ;LOC. 0-1
                                        NEG     @(R0)+          ;TRY NEGATE   LOC. 0--1  R0-402
                                        BPL     NEG30           ;CC-1001?
                                        BEQ     NEG30
                                        BVS     NEG30
                                        BCS     NEG31
                                NEG30:
                                        EMT                     ;NEG DID NOT SET CC'S CORRECTLY
                                NEG31:  INC     (R4)            ;LOC. 0-0
                                        BEQ     NEG32
                                        EMT                     ;DATA RESULT OF NEG INCORRECT
                                NEG32:  COMB    @#1             ;LOC 0-177400
                                        INC     @#0             ;LOC. 0-177401
                                        NEGB    @(R0)+          ;TRY NEGB LOC. 0-177777  R0-404
                                        BMI     NEG33
                                        EMT                     ;NEGB FAILED WITH EVEN BYTE
                                NEG33:  NEGB    @(R0)+          ;TRY NEGB  LOC.0-777  R0-406
                                        BPL     NEG34
                                        EMT                     ;NEGB FAILED WITH ODD BYTE
                                NEG34:  COMB    @#1             ;LOC. 0-177377
                                        INCB    @#1             ;LOC. 0-177777
                                        INC     (R4)            ;LOC. 0-0
                                        BEQ     TS66
                                        EMT                     ;DATA RESULT OF NEGB'S INCORRECT
                                ;*****************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 4 SINGLE OPERAND INSTRUCTIONS.
                                ;R0 IS SET TO 400. A CLR INSTRUCTION IS EXECUTED IN MODE 4 TO CLEAR
                                ;LOC. 376. R0 IS RESET TO 400 AND A COM INSTRUCTION USING MODE 4
                                ;COMPLEMENTS LOC.376.
                                ;       TWO INC INSTRUCTIONS AND A MODE 4 INSTRUCTION ARE EXECUTED
                                ;TO COMPLETE THE TEST.
                                ;*****************************************************************************
                                ;TEST 66        TEST MODE 4 USING SOP INSTS
                                ;*****************************************************************************
                                TS66:
                                        CLR     R0              ;SET R0-400
                                        COMB    R0
                                        INC     R0
```

```
7383  005506  005040                    CLR     -(R0)           ;TRY TO CLEAR USING MODE 4
7384  005510  001401                    BEQ     SOP4A
 (2)  005512  104000                    EMT                     ;CLR DID NOT SET Z-BIT
7385  005514  005200            SOP4A:  INC     R0              ;RESET R0
7386  005516  005200                    INC     R0
7387  005520  005140                    COM     -(R0)           ;TRY TO COMPLEMENT USING MODE 4
7388  005522  100004                    BPL     SOP4B
7389  005524  005200                    INC     R0              ;MOVE POINTER
7390  005526  005200                    INC     R0
7391  005530  005240                    INC     -(R0)
7392  005532  001401                    BEQ     TS67
 (2)  005534                    SOP4B:
 (3)  005534  104000                    EMT                     ;CHECK CUMMULATIVE RESULT OF ABOVE INST.
7393
7394                                    ;*****************************************************************************
7395                                    ;
7396                                    ;       THIS TEST VERIFIES MODE 5 SINGLE OPERAND INSTRUCTIONS.  IT
7397                                    ;USES LOCATION 0 AS ITS TARGET DATA.  A TABLE LOCATED AT LOC. 372
7398                                    ;THRU 374 IS USED TO SUPPLY THE ADDRESS OF LOCATION 0 TO THE
7399                                    ;INSTRUCTIONS UNDER TEST.
7400                                    ;       R0 IS SET TO 376, (THE START OF THE ADDRESS TABLE) +2,
7401                                    ;AND A CLR INSTRUCTION IS EXECUTED WITH MODE 3 TO CLEAR
7402                                    ;LOC. 0.  THEN R0 IS INCREMENTED BY TWO AND TWO OTHER MODE 3
7403                                    ;INSTRUCTIONS OPERATE ON LOC. 0 TO VERIFY THE DATA RESULTS OF
7404                                    ;THE TEST.  THE PROPER DECREMENTING OF THE REGISTER IS ALSO
7405                                    ;VERIFIED IN THIS MANNER.
7406                                    ;       IF A FAILURE IS DETECTED BE SURE TO VERIFY THAT THE TABLE
7407                                    ;(LOC. 372 THRU 374) HAS THE PROPER VALUES (0).
7408                                    ;
7409                                    ;*****************************************************************************
 (2)                                    ;TEST 67        TEST MODE 5 USING SOP INSTS
 (3)                                    ;*****************************************************************************
 (2)  005536                    TS67:
7410  005536  012700  000370           MOV     #370,R0         ;CLEAR LOCATION 370-376
7411  005542  005020                    CLR     (R0)+           ;370
7412  005544  005020                    CLR     (R0)+           ;372
7413  005546  005020                    CLR     (R0)+           ;374
7414  005550  005010                    CLR     (R0)            ;376
7415  005552  005000                    CLR     R0              ;SET R0=376 (LOW BYTE)
7416  005554  005020                    CLR     (R0)+
7417  005556  105400                    NEGB    R0
7418  005560  005050                    CLR     @-(R0)          ;TRY TO CLEAR LOC 0 W/MODE 5
7419  005562  001401                    BEQ     SOP5A
 (2)  005564  104000                    EMT                     ;CLR DID NOT SET Z-BIT
7420  005566  005200            SOP5A:  INC     R0              ;RESET R0
7421  005570  005200                    INC     R0
7422  005572  005150                    COM     @-(R0)          ;TRY TO COMPLEMENT LOC. 0 W/MODE 5
7423  005574  100002                    BPL     SOP5B
7424  005576  005250                    INC     @-(R0)          ;TRY TO INCREMENT LOC. 0 W/MODE 5
7425  005600  001401                    BEQ     TS70
 (2)  005602                    SOP5B:
 (3)  005602  104000                    EMT                     ;TEST CUMMULATIVE RESULT OF ABOVE INSTS
7426
7427                                    ;*****************************************************************************
7428                                    ;
7429                                    ;       THIS TEST VERIFIES MODE 6 SINGLE OPERAND INSTRUCTIONS.  IT
```

```
                                ;USES LOCATION 0 AS ITS TARGET DATA.  R0 IS SET TO 400 USING
                                ;PREVIOUSLY TESTED INSTRUCTIONS AND A MODE 6 CLR INSTRUCTION IS
                                ;EXECUTED ON LOC. 0 USING R0 AND A -400 OFFSET.  COM AND INC
                                ;INSTRUCTIONS ARE THEN USED TO VERIFY THE DATA.
                                ;
                                ;*****************************************************************************
                                ;TEST 70        TEST MODE 6 USING SOP INSTS
                                ;*****************************************************************************
(2)  005604
7436  005604  005000            TS70:   CLR     R0              ;SET R0=400
7437  005606  105100                    COMB    R0
7438  005610  005200                    INC     R0
7439  005612  005060  177400            CLR     -400(R0)        ;TRY TO CLEAR LOCATION 0 W/MODE 6
7440  005616  001401                    BEQ     SOP6A
(2)  005620  104000                     EMT                     ;CLR DID NOT SET Z-BIT
7441  005622  005160  177400    SOP6A:  COM     -400(R0)        ;TRY TO COMPLEMENT LOCATION 0 W/MODE 6
7442  005626  100003                    BPL     SOP6B
7443  005630  005260  177400            INC     -400(R0)        ;TRY TO INCREMENT LOCATION 0 W/MODE 6
7444  005634  001401                    BEQ     TS71
(2)  005636             SOP6B:
(3)  005636  104000                     EMT                     ;TEST CUMMULATIVE RESULT OF ABOVE INSTS
7445
                                ;*****************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 7 SINGLE OPERAND INSTRUCTIONS.  IT USES
                                ;THE POINTER TO LOC. 0 WHICH IS STORED AT LOC. 402.
                                ;       R0 IS SET TO 400 AND A MODE 7 CLR INSTRUCTION IS
                                ;EXECUTED WITH A +2 OFFSET TO CLEAR LOC. 0.
                                ;       SEVERAL OTHER MODE 7 INSTRUCTIONS ARE THEN USED ON THE COMMON
                                ;LOCATION TO VERIFY THE DATA RESULTS.
                                ;
                                ;*****************************************************************************
                                ;TEST 71        TEST MODE 7 USING SOP INST.
                                ;*****************************************************************************
(2)  005640
7456  005640  005000            TS71:   CLR     R0              ;SET R0=400
7457  005642  105100                    COMB    R0
7458  005644  005200                    INC     R0
7459  005646  005210                    INC     (R0)            ;R0=1
7460  005650  005070  000002            CLR     @2(R0)          ;TRY TO CLEAR LOC. 0 W/MODE 7
7461  005654  001401                    BEQ     SOP7A
(2)  005656  104000                     EMT                     ;CLR DID NOT SET Z-BIT
7462  005660  005170  000002    SOP7A:  COM     @2(R0)          ;TRY TO COMPLEMENT LOC. 0 W/MODE 7
7463  005664  100003                    BPL     SOP7B
7464  005666  005270  000002            INC     @2(R0)          ;TRY TO INCREMENT LOC. 0 W/MODE 7
7465  005672  001401                    BEQ     TS72
(2)  005674             SOP7B:
(3)  005674  104000                     EMT                     ;TEST CUMMULATIVE RESULT OF ABOVE INSTS.
7466
7467                            ;*****************************************************************************
                                ;TEST 72        TEST MODE 4 WITH NEGATE INSTRUCTION
                                ;*****************************************************************************
(2)  005676
7468  005676  005000            TS72:   CLR     R0
7469  005700  005010                    CLR     (R0)
7470  005702  005120                    COM     (R0)+           ;LOC. 0=177777,  R0=2
```

```
 7471  005704  005440                NEG     -(R0)           ;TRY NEGATE, LOC. 0=1
 7472  005706  100403                BMI     NEG40           ;CC=0001?
 7473  005710  001402                BEQ     NEG40
 7474  005712  102401                BVS     NEG40
 7475  005714  103401                BCS     NEG41
  (1)  005716                NEG40:
  (2)  005716  104000                EMT                     ;NEG DID NOT SET CC'S CORRECTLY
 7476  005720  005400        NEG41:  NEG     R0              ;TST R0 WITH A NEG.
 7477  005722  001401                BEQ     NEG42
  (2)  005724  104000                EMT                     ;R0 NOT DECREMENTED PROPERLY
 7478  005726  005310        NEG42:  DEC     (R0)            ;TEST DTA RESULT OF NEG
 7479  005730  001401                BEQ     TS73
  (3)  005732  104000                EMT                     ;DATA RESULT OF NEG INCORRECT
 7480                       ;*****************************************************************************
  (2)                       ;TEST 73        TEST MODE 5 WITH NEGATE INSTRUCTION
  (3)                       ;*****************************************************************************
  (2)  005734                TS73:
 7481  005734  005000                CLR     R0              ;R0=0
 7482  005736  005010                CLR     (R0)            ;LOC. 0=0
 7483  005740  105100                COMB    R0              ;R0=377
 7484  005742  005200                INC     R0              ;R0=400
 7485  005744  005010                CLR     (R0)            ;SET 400 = 0
 7486  005746  005004                CLR     R4              ;R4=0
 7487  005750  005314                DEC     (R4)            ;LOC. 0=177777
 7488  005752  005450                NEG     @-(R0)          ;TRY NEGATE:  LOC. 0=1
 7489  005754  100403                BMI     NEG50           ;CC=0001?
 7490  005756  001402                BEQ     NEG50
 7491  005760  102401                BVS     NEG50
 7492  005762  103401                BCS     NEG51
  (1)  005764                NEG50:
  (2)  005764  104000                EMT                     ;NEG DID NOT SET CC'S CORRECTLY
 7493  005766  005314        NEG51:  DEC     (R4)
 7494  005770  001401                BEQ     NEG52
  (2)  005772  104000                EMT                     ;DATA RESULT OF NEG INCORRECT
 7495  005774  105100        NEG52:  COMB    R0
 7496  005776  005300                DEC     R0
 7497  006000  001401                BEQ     TS74
  (3)  006002  104000                EMT                     ;REGISTER NOT DECREMENTED PROPERLY
 7498                       ;*****************************************************************************
  (2)                       ;TEST 74        TEST MODE 6 WITH NEGATE
  (3)                       ;*****************************************************************************
  (2)  006004                TS74:
 7499  006004  005000                CLR     R0              ;R0=0
 7500  006006  005004                CLR     R4              ;R4=0
 7501  006010  105100                COMB    R0              ;R0=377
 7502  006012  005014                CLR     (R4)            ;LOC. 0=0
 7503  006014  105024                CLRB    (R4)+           ;LOC. 0=177777,  R4=1
 7504  006016  105114                COMB    (R4)            ;LOC. 0=177400
 7505  006020  005460  177401        NEG     -377(R0)        ;LOC. 0=400
 7506  006024  100403                BMI     NEG60           ;CC=0001
 7507  006026  001402                BEQ     NEG60
 7508  006030  102401                BVS     NEG60
 7509  006032  103401                BCS     NEG61
  (1)  006034                NEG60:
  (2)  006034  104000                EMT                     ;NEG DID NOT SET CC'S CORRECTLY
 7510  006036  105314        NEG61:  DECB    (R4)
```

```
                                        BEQ     TS75
                                        EMT                     ;DATA RESULT OF NEG INCORRECT
                                ;*******************************************************************************
                                ;TEST 75        TEST MODE 7 W/ NEGATE
                                ;*******************************  ************************************************
                                TS75:
                                        CLR     R0              ;R0=0
                                        CLR     (R0)            ;LOC. 0=0
                                        COM     (R0)            ;LOC. 0=177777
                                        COMB    R0              ;R0=377
                                        NEGB    @5(R0)          ;R0+5=404,  404=1,  LOC. 0=777
                                        BMI     NEG70           ;CC=0001?
                                        BEQ     NEG70
                                        BVS     NEG70
                                        BCS     NEG71
                                NEG70:
                                        EMT                     ;NEG DID NOT SET CC'S CORRECTLY
                                NEG71:  COMB    R0              ;R0=0
                                        COMB    (R0)+           ;LOC. 0=400,  R0=1
                                        DECB    (R0)            ;LOC. 0=0
                                        NEG     0               ;USE NEG MODE 67 TO TST FOR ZERO
                                        BEQ     TS76
                                        EMT                     ;DATA RESULT OF NEG WAS INCORRECT

                                ;*******************************************************************************
                                ;
                                ;       THIS TEST VERIFIES PROGRAM COUNTER ADDRESSING WITH SOP
                                ;INSTRUCTIONS.  CLR MODE 77 IS USED TO CLEAR THE LOCATION FOLLOWING THE
                                ;INSTRUCTION (SOPX).  THEN SINGLE OPERAND INSTRUCTIONS WITH MODES 37, 67, AND
                                ;77, USING INDIRECT POINTER SOPXAD ARE USED TO VERIFY THE DATA RESULTS
                                ;OF THESE INSTRUCTIONS.
                                ;
                                ;*******************************************************************************
                                ;TEST 76        TEST SOP INSTRUCTIONS MODES 2,3,6,7 WITH REGISTER 7
                                ;*******************************************************************************
                                TS76:
                                        CLR     (R7)+           ;CLEAR NEXT LOCATION: (SOPX)
                                SOPX:   -1                      ;USE MODE 27
                                        BEQ     SOPA
                                        EMT                     ;CLR DID NOT SET Z-BIT
                                SOPA:   INC     @#SOPX          ;INC SOPX W/MODE 37
                                        NEG     SOPX            ;NEGATE SOPX W/MODE 67
                                        BPL     SOPB
                                        INC     @SOPXAD         ;INC SOPX W/MODE 77
                                        BEQ     TS77
                                SOPB:
                                        EMT                     ;INC DID NOT SET Z-BIT
                                SOPXAD: SOPX                    ;INDIRECT ADDRESS OF SOPX

                                ;*******************************************************************************
                                ;
                                ;       THIS TEST VERIFIES SINGLE OPERAND NON-MODIFYING INSTRUCTIONS
                                ;USING MODE 0.  R0 IS SET TO ZERO AND THE CONDITION CODES ARE SET
                                ;TO THE COMPLEMENT OF THAT EXPECTED BY THE INSTRUCTION.  A TST INSTRUCTION
                                ;IS EXECUTED AND CONDITIONAL BRANCHES ARE USED TO TEST THE CONDITION
                                ;CODES.
```

```
                                ;
                                ;*******************************************************************************
                                ;TEST 77         TEST MODE 0 SOP NON-MODIFYING
                                ;*******************************************************************************
                                TS77:
 (2) 006144
7554 006144  005000                    CLR     R0              ;INITIALIZE R0=0
7555 006146  000277                    SCC                     ;SET CC=1011
 (2) 006150  000244                    CLZ
 (3) 006152  005700                    TST     R0              ;TRY TST W/ MODE 0
 (2) 006154  102403                    BVS     SNM0A           ;CHECK THAT CC=0100
 (3) 006156  100402                    BMI     SNM0A
 (2) 006160  103401                    BCS     SNM0A
 (3) 006162  001401                    BEQ     TS100
                                SNM0A:
 (2) 006164
 (3) 006164  104000                    EMT                     ;CONDITION CODES NOT SET PROPERLY
                                ;*******************************************************************************
                                ;
                                ;       THIS TEST VERIFIES SINGLE OPERAND NON-MODIFYING BYTE INSTRUCTIONS WITH MODE 0.
                                ;R0 IS SET TO 377 AND COMPLEMENT OF THE EXPECTED CONDITION CODES
                                ;IS LOADED IN PSW.  A TSTB INSTRUCTION IS EXECUTED AND THE RESULTS
                                ;ARE CHECKED WITH SEVERAL CONDITIONAL BRANCH INSTRUCTIONS.
                                ;       THIS VERIFIES THAT THE PROPER BYTE WAS TESTED.
                                ;
                                ;*******************************************************************************
                                ;TEST 100        TEST MODE 0 EVEN BYTE W/ SOP NON-MODIFYING
                                ;*******************************************************************************
                                TS100:
 (2) 006166
7574 006166  005000                    CLR     R0              ;INITIALIZE
7575 006170  105100                    COMB    R0              ;R0=377
7576 006172  000277                    SCC                     ;SET CC=0111
7577 006174  000250                    CLN
7578 006176  105700                    TSTB    R0              ;TRY TST EVEN BYTE
7579 006200  102402                    BVS     SNMB0A          ;CHECK CC=1000
7580 006202  101401                    BLOS    SNMB0A
7581 006204  100401                    BMI     TS101
 (2) 006206
 (3) 006206  104000             SNMB0A: EMT                    ;CONDITION CODES NOT SET PROPERLY
                                ;*******************************************************************************
                                ;
                                ;       THIS TEST VERIFIES SINGLE OPERAND INSTRUCTIONS WITH MODE 1.
                                ;R0 IS USED TO POINT TO AND CLEAR LOC. 0.  THE COMPLEMENT OF THE
                                ;EXPECTED CONDITION CODES ARE LOADED IN THE PSW.  A TST INSTRUCTION
                                ;IS THEN EXECUTED ON LOC. 0 USING R0 AND CONDITIONAL BRANCHES TEST
                                ;THE RESULTS.
                                ;
                                ;*******************************************************************************
                                ;TEST 101        TEST MODE 1 SOP NON-MODIFYING
                                ;*******************************************************************************
                                TS101:
 (2) 006210
7592 006210  005000                    CLR     R0              ;POINT TO LOC 0
7593 006212  005010                    CLR     (R0)            ;CLEAR LOC 0
7594 006214  000277                    SCC                     ;INITIALIZE
7595 006216  000244                    CLZ                     ;CC=1011
7596 006220  005710                    TST     (R0)            ;TRY TST W/ MODE 1
```

```
                                        BVS     SNM1A           ;CHECK CC=0100
                                        BCS     SNM1A
                                        BMI     SNM1A
                                        BEQ     TS102
                                SNM1A:
                                        EMT                     ;CC'S NOT SET PROPERLY

                                ;*******************************************************************************
                                ;
                                ;       THIS TEST SETS LOCATION 0 TO 377 AND THEN USES R0 TO TEST
                                ;THE EVEN BYTE AND THE ODD BYTE USING SOP BYTE INSTRUCTIONS WITH MODE 1.
                                ;AGAIN, CONDITIONAL BRANCHES ARE USED TO VERIFY THE SETTING OF THE
                                ;PROPER CONDITION CODE BITS.
                                ;
                                ;*******************************************************************************
                                ;TEST 102        TEST MODE 1 BYTE INST. NON-MODIFYING
                                ;*******************************************************************************
                                TS102:
                                        CLR     R0              ;POINT TO LOC 0
                                        CLR     (R0)            ;CLEAR LOC 0
                                        COMB    (R0)            ;COMPLEMENT BYTE 0
                                        SCC                     ;SET CC=0111
                                        CLN
                                        TSTB    (R0)            ;TRY TST ON EVEN BYTE
                                        BVS     SNMB1A
                                        BLOS    SNMB1A
                                        BMI     SNMB1B
                                SNMB1A:
                                        EMT                     ;CC'S NOT CORRECT
                                SNMB1B: CLR     R0
                                        INC     R0
                                        SCC                     ;SET CC=1011
                                        CLZ
                                        TSTB    (R0)            ;TRY TO TST AN ODD BYTE
                                        BVS     SNMB1C          ;CHECK CC=0100
                                        BCS     SNMB1C
                                        BMI     SNMB1C
                                        BEQ     TS103
                                SNMB1C:
                                        EMT                     ;CC'S NOT CORRECT

                                ;*******************************************************************************
                                ;
                                ;       THIS TEST VERIFIES THE SINGLE-OPERAND NON-MODIFYING INSTRUCTIONS
                                ;USING MODE 2.  IT USES THE IDENTICAL PROCEDURE EMPLOYED IN THE
                                ;MODE 1 TESTS.  ADDITIONALLY, THE REGISTER IS CHECKED TO ASSURE THAT
                                ;IT IS INCREMENTED PROPERLY.
                                ;
                                ;*******************************************************************************
                                ;TEST 103        TEST MODE 2 WITH SOP NON-MODIFYING
                                ;*******************************************************************************
                                TS103:
                                        CLR     R0              ;INITIALIZE R0=0
                                        CLR     (R0)            ;CLEAR LOC 0
                                        SCC                     ;SET CC=1011
                                        CLZ
```

```
                                        TST     (R0)+           ;TRY TST W/ MODE 2
                                        BVS     SNM2A           ;CHECK CC=0100
                                        BCS     SNM2A
                                        BMI     SNM2A
                                        BEQ     SNM2B
                                SNM2A:
                                        EMT                     ;CC'S NOT CORRECT
                                SNM2B:  DEC     R0              ;RESET R0
                                        DEC     R0
                                        BEQ     TS104
                                        EMT                     ;MODE 2 DID NOT INC REG CORRECTLY

                                ;*******************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 2 SINGLE OPERAND NON-MODIFYING BYTE
                                ;INSTRUCTIONS IT USES R0 TO POINT TO LOC. 0.  WITH LOCATION 0
                                ;SET TO 377, THE EVEN AND ODD BYTE IS TESTED WITH TSTB INSTRUCTIONS
                                ;TO VERIFY THE CORRECT CC ARE SET.  THE REGISTER IS CHECKED FOR
                                ;PROPER INCREMENTING.
                                ;
                                ;*******************************************************************************
                                ;TEST 104       TEST MODE 2 - BYTE W/ SOP NON-MODIFYING
                                ;*******************************************************************************
                                TS104:
                                        CLR     R0              ;CLEAR R0
                                        CLR     (R0)            ;CLEAR LOC 0
                                        COMB    (R0)            ;SET LOC 0=377
                                        SCC                     ;SET CC=0111
                                        CLN
                                        TSTB    (R0)+           ;TRY TST OF EVEN BYTE
                                        BVS     SNMB2A
                                        BLOS    SNMB2A
                                        BMI     SNMB2B
                                SNMB2A:
                                        EMT                     ;CC'S NOT SET CORRECTLY
                                SNMB2B: DEC     R0              ;DECREMENT R0
                                        BEQ     SNMB2C
                                        EMT                     ;MODE 2 DID NOT INC REG CORRECTLY
                                SNMB2C: INC     R0              ;POINT TO ODD BYTE
                                        SCC                     ;SET CC=1011
                                        CLZ
                                        TSTB    (R0)+           ;TRY TST OF ODD BYTE
                                        BVS     SNMB2D          ;CHECK CC'S=0100
                                        BCS     SNMB2D
                                        BMI     SNMB2D
                                        BEQ     SNMB2E
                                SNMB2D:
                                        EMT                     ;CC'S NOT CORRECT
                                SNMB2E: DEC     R0
                                        DEC     R0
                                        BEQ     TS105
                                        EMT                     ;R0 DID NOT INCREMENT PROPERLY

                                ;*******************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 3 SINGLE OPERAND NON-MODIFYING INSTRUCTIONS.
```

```
                                ;A POINTER IN A TABLE AT LOC. 376 IS USED TO TEST LOCATION 0.
                                ;THE CC'S AND THE REGISTER ARE CHECKED FOLLOWING THE
                                ;TST MODE 3 INSTRUCTION.
                                ;
                                ;*****************************************************************************
                                ;TEST 105       TEST MODE 3 W/ SOP NON-MODIFYING INSTS
                                ;*****************************************************************************
                                TS105:
006424  005000                          CLR     R0              ;R0=0
006426  005010                          CLR     (R0)            ;CLEAR LOC 0
006430  105100                          COMB    R0              ;R0=376
006432  005300                          DEC     R0
006434  000277                          SCC                     ;SET CC=1011
006436  000244                          CLZ
006440  005730                          TST     @(R0)+          ;TRY TST W/ MODE 3
006442  102403                          BVS     SNM3A           ;CHECK CC=0100
006444  103402                          BCS     SNM3A
006446  100401                          BMI     SNM3A
006450  001401                          BEQ     SNM3B
006452                          SNM3A:
006452  104000                          EMT                     ;CC'S NOT CORRECT
006454  005300                  SNM3B:  DEC     R0              ;R0=377
006456  105100                          COMB    R0              ;R0=0
006460  001401                          BEQ     TS106
006462  104000                          EMT                     ;MODE 3 DID NOT INC REG CORRECTLY
                                ;*****************************************************************************
                                ;
                                ;       THIS TEST VERIFIES SOP NON-MODIFYING BYTE INSTRUCTIONS MODE 3
                                ;LOC. 0 IS SET TO 377.  TABLE AT LOC. 402-404 IS USED TO TEST
                                ;BYTE 0 AND BYTE 1.  THE REGISTER IS CHECKED FOR PROPER INCREMENTING AND
                                ;THE CC'S ARE VERIFIED.
                                ;       THE TABLE AT LOC. 402-404 SHOULD CONTAIN 0 AND 1 BEFORE AND
                                ;AFTER THE TEST IS RUN.
                                ;
                                ;*****************************************************************************
                                ;TEST 106       TEST MODE 3 - BYTES W/ SOP NON-MODIFYING INSTS.
                                ;*****************************************************************************
                                TS106:
006464  005000                          CLR     R0              ;R0=0
006466  005010                          CLR     (R0)            ;CLEAR LOC 0
006470  105110                          COMB    (R0)            ;LOC. 0 =377
006472  105100                          COMB    R0
006474  005200                          INC     R0
006476  005720                          TST     (R0)+           ;R0=402
006500  000277                          SCC                     ;CC=0111
006502  000250                          CLN
006504  105730                          TSTB    @(R0)+          ;TRY TST OF EVEN BYTE
006506  102402                          BVS     SNMB3A          ;CHECK CC=1000
006510  101401                          BLOS    SNMB3A
006512  100401                          BMI     SNMB3B
006514                          SNMB3A:
006514  104000                          EMT                     ;CC'S NOT CORRECT
006516  000277                  SNMB3B: SCC                     ;SET CC=1011
006520  000244                          CLZ
006522  105730                          TSTB    @(R0)+          ;TRY TST OF ODD BYTE
```

```
                                        BVS     SNM83C          ;CHECK CC=0100
                                        BCS     SNM83C
                                        BMI     SNM83C
                                        BEQ     SNM83D
                                SNM83C:
                                        EMT                     ;CC'S NOT CORRECT
                                SNM83D: TST     (R0)+           ;R0=410
                                        TST     (R0)
                                        BMI     TS107
                                        EMT                     ;TSTB DID NOT INCREMENT R0 CORRETCLY
                                ;*******************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 4 SOP NON-MODIFYING INSTRUCTIONS.
                                ;LOC. 0 IS SET TO -1 AND THE CC'S ARE SET TO THE COMPLEMENT OF THE
                                ;EXPECTED RESULTS.  R0 AND SET TO 2 AND A TST MODE 4 IS EXECUTED.
                                ;THE CC'S ARE CHECKED WITH CONDITIONAL BRANCH INSTRUCTIONS AND THE REGISTER
                                ;IS CHECKED FOR PROPER DECREMENTING.
                                ;
                                ;*******************************************************************************
                                ;TEST 107        TEST MODE 4 W/ SOP NON-MODIFYING INSTS
                                ;*******************************************************************************
                                TS107:
                                        CLR     R0              ;R0=0
                                        CLR     (R0)            ;LOC 0=0
                                        COM     (R0)+           ;LOC 0=-1
                                        SCC                     ;SET CC=1011
                                        CLZ
                                        TST     -(R0)           ;TRY TST W/ MODE 4
                                        BVS     SNM4A           ;CHECK CC=0100
                                        BLOS    SNM4A
                                        BMI     SNM4B
                                SNM4A:
                                        EMT                     ;CC'S NOT CORRECT
                                SNM4B:  TST     R0
                                        BEQ     TS110
                                        EMT                     ;TST MODE 4 DID NOT DEC R0 CORRECTLY
                                ;*******************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 5 SOP NON-MODIFYING INSTRUCTIONS.
                                ;IT USES A POINTER AT LOC. 376 TO TEST LOC. 0.  R0 IS SET
                                ;TO 400, A TST MODE 5 INSTRUCTION IS EXECUTED AND THE CC'S CHECKED.
                                ;R0 IS CHECKED TO INSURE PROPER DECREMENTING.
                                ;
                                ;*******************************************************************************
                                ;TEST 110        TEST MODE 5 W/ SOP NON-MODIFYING INSTS
                                ;*******************************************************************************
                                TS110:
                                        CLR     R0              ;R0=0
                                        CLR     (R0)            ;LOC 0=0
                                        COM     (R0)            ;LOC 0=-1
                                        COMB    R0              ;R0=377
                                        INC     R0              ;R0=400
                                        SCC                     ;SET CC=0111
                                        CLN
                                        TST     @-(R0)          ;TRY TST W/ MODE 5
```

```
  7774  006620  102402                    BVS     SNM5A           ;CHECK CC=1000
  7775  006622  101401                    BLOS    SNM5A
  7776  006624  100401                    BMI     SNM5B
   (1)  006626                    SNM5A:
   (2)  006626  104000                    EMT                     ;CC'S NOT SET PROPERLY
  7777  006630  005200            SNM5B:  INC     R0              ;R0=377
  7778  006632  105100                    COMB    R0              ;R0=0
  7779  006634  001401                    BEQ     TS111
   (3)  006636  104000                    EMT                     ;MODE 5 DID NOT DEC R0 CORRECTLY
  7780
  7781                            ;*****************************************************************************
  7782                            ;
  7783                            ;       THIS TEST VERIFIES MODE 6 SOP NON-MODIFYING INSTRUCTIONS.
  7784                            ;R0 IS SET TO 377 AND A MODE 6 TST INSTRUCTION IS EXECUTED
  7785                            ;USING R0 AND AN OFFSET OF -377.  THE CC'S ARE CHECKED AS WELL
  7786                            ;AS R0 TO INSURE IT WAS NOT ALTERED.
  7787                            ;
  7788                            ;*****************************************************************************
   (2)                            ;TEST 111       TEST MODE 6 W/ SOP NON-MODIFYING INSTS
   (3)                            ;*****************************************************************************
   (2)  006640                    TS111:
  7789  006640  005000                    CLR     R0              ;R0=0
  7790  006642  005010                    CLR     (R0)            ;LOC 0=0
  7791  006644  005110                    COM     (R0)            ;LOC 0=-1
  7792  006646  105100                    COMB    R0              ;R0=377
  7793  006650  000277                    SCC                     ;SET CC=0111
  7794  006652  000250                    CLN
  7795  006654  005760  177401            TST     -377(R0)        ;TRY TST W/ MODE 6
  7796  006660  102402                    BVS     SNM6A           ;CHECK CC=1000
  7797  006662  101401                    BLOS    SNM6A
  7798  006664  100401                    BMI     SNM6B
   (1)  006666                    SNM6A:
   (2)  006666  104000                    EMT                     ;CC'S INCORRECT
  7799  006670  105100            SNM6B:  COMB    R0              ;R0=0
  7800  006672  001401                    BEQ     TS112
   (3)  006674  104000                    EMT                     ;TST MODE 6 INCORRECTLY CHANGED R0
```

```
                                        ;*******************************************************************************
                                        ;
                                        ;       THIS TEST VERIFIES MODE 7 SOP NON-MODIFYING INSTRUCTIONS.
                                        ;IT USES A POINTER TO LOC. 0 STORED AT LOC. 400 TO TST LOC. 0.
                                        ;R0 IS SET TO 377 AND LOC. 0 IS TESTED THRU THE POINTER AT 400 USING
                                        ;R0 AND AN OFFSET OF 1.
                                        ;
                                        ;*******************************************************************************
                                        ;TEST 112       TEST MODE 7 W/ SOP NON-MODIFYING INSTS.
                                        ;*******************************************************************************
006676                                  TS112:
006676  005000                                  CLR     R0              ;R0=0
006700  005010                                  CLR     (R0)            ;LOC 0=0
006702  005110                                  COM     (R0)            ;LOC 0=-1
006704  105100                                  COMB    R0              ;R0=377
006706  000277                                  SCC                     ;CC=0111
006710  000250                                  CLN
006712  005770  000001                          TST     @1(R0)          ;TRY TST W/ MODE 7
006716  102402                                  BVS     SNM7A           ;CHECK CC=1000
006720  101401                                  BLOS    SNM7A
006722  100401                                  BMI     SNM7B
006724                                  SNM7A:
006724  104000                                  EMT                     ;CC'S NOT CORRECT
006726  105100                          SNM7B:  COMB    R0              ;R0=0
006730  001401                                  BEQ     TS113
006732  104000                                  EMT                     ;TST MODE 7 INCORRECTLY CHANGED R0

                                        ;*******************************************************************************
                                        ;
                                        ;       THIS TEST VERIFIES MODE 0 DOUBLE OPERAND INSTRUCTIONS.  IT SETS
                                        ;DATA IN R0 AND R4 AND USES THE ADD INSTRUCTION TO TEST THE DOP
                                        ;MICROCODE.
                                        ;
                                        ;*******************************************************************************
                                        ;TEST 113       TEST MODE 0 DOUBLE-OPERAND (DOP) INSTS.
                                        ;*******************************************************************************
006734                                  TS113:
006734  005000                                  CLR     R0              ;R0=0
006736  005100                                  COM     R0              ;R0=-1
006740  005004                                  CLR     R4              ;R4=0
006742  060004                                  ADD     R0,R4           ;TRY ADD: R4=-1
006744  005204                                  INC     R4              ;R4=0
006746  001401                                  BEQ     TS114
006750  104000                                  EMT                     ;ADD INST. FAILED W/ MODE 0

                                        ;*******************************************************************************
                                        ;
                                        ;       THIS TEST VERIFIES THE MOVE INSTRUCTION WITH MODE 0 TO MODE 0.
                                        ;
                                        ;*******************************************************************************
                                        ;TEST 114       MOV MODE 0 TO MODE 0
                                        ;*******************************************************************************
006752                                  TS114:
006752  005000                                  CLR     R0              ;R0=0
006754  005004                                  CLR     R4              ;R4=0
```

```
                                        COM     R0              ;R0=-1
                                        MOV     R0,R4           ;TRY MOVE -1 TO R4
                                        INC     R4              ;INC R4
                                        BEQ     TS115
                                        EMT                     ;MOVE FAILED MODE 0 TO MODE 0

                                ;*******************************************************************************
                                ;
                                ;       THIS TEST VERIFIES THE SUBTRACT INSTRUCTION WITH MODE 0,0.
                                ;
                                ;*******************************************************************************
                                ;TEST 115       TEST SUB MODE 0,0
                                ;*******************************************************************************
                                TS115:
                                        CLR     R0              ;R0=0
                                        CLR     R4              ;R4=0
                                        INC     R4              ;R4=1
                                        SUB     R4,R0           ;TRY SUB 0,0   R0=-1
                                        BPL     SUB0            ;CC=1001
                                        BEQ     SUB0
                                        BVS     SUB0
                                        BCS     SUB0A
                                SUB0:
                                        EMT                     ;CONDITION CODE FAILED ON SUB
                                SUB0A:  INC     R0
                                        BEQ     TS116
                                        EMT                     ;DATA RESULT OF SUB FAILED

                                ;*******************************************************************************
                                ;
                                ;       THIS TEST QUICKLY VERIFIES THE REMAINING DOP MODIFYING INSTRUCTIONS
                                ;WITH MODE 0,0 TO PROVIDE A BASELINE FOR SUBSEQUENT TESTS.
                                ;SINGLE OPERAND INSTRUCTIONS ARE USED TO SET UP DATA IN R0 AND R4
                                ;BEFORE EACH OF THE SEVERAL DOP MODIFYING INSTRUCTIONS ARE USED AND
                                ;VERIFIED.
                                ;
                                ;*******************************************************************************
                                ;TEST 116       TEST ALL THE DOP INSTRUCTIONS W/ SOURCE MODE 0,0
                                ;*******************************************************************************
                                TS116:
                                        CLR     R0              ;R0=0
                                        MOV     R0,R4           ;TRY MOVE MODE 0,0
                                        BEQ     DOP0A
                                        EMT                     ;Z-BIT NOT SET
                                DOP0A:  INC     R0              ;R0=1
                                        COM     R0              ;R0=177776
                                        COM     R4              ;R4=177777
                                        BIC     R0,R4           ;TRY BIC: R4=1
                                        DEC     R4              ;R4=0
                                        BEQ     DOP0B
                                        EMT                     ;BIC CLEAR RESULT INCORRECT
                                DOP0B:  BIS     R0,R4           ;TRY BIS: R4=177777
                                        INC     R4
                                        INC     R4              ;R4=0
                                        BEQ     DOP0C
                                        EMT                     ;RESULT OF BIS INCORRECT
```

```
 7888  007060  005000                DOP0C:  CLR     R0              ;R0=0
 7889  007062  105100                        COMB    R0              ;R0=377
 7890  007064  005004                        CLR     R4              ;R4=0
 7891  007066  005104                        COM     R4              ;R4=177777
 7892  007070  040004                        BIC     R0,R4           ;R4=177400
 7893  007072  060004                        ADD     R0,R4           ;TRY ADD: R4=177777
 7894  007074  005204                        INC     R4              ;R4=0
 7895  007076  001401                        BEQ     DOP0D
  (2)  007100  104000                        EMT                     ;RESULT OF ADD INCORRECT
 7896  007102  160004                DOP0D:  SUB     R0,R4           ;177401=R4
 7897  007104  105404                        NEGB    R4              ;R4=177777
 7898  007106  005204                        INC     R4              ;RD=0
 7899  007110  001401                        BEQ     TS117
  (3)  007112  104000                        EMT                     ;RESULT OF SUB INCORRECT
 7900
 7901                                ;*****************************************************************************
 7902                                ;
 7903                                ;       THIS TEST VERIFIES MODE 0,X DOUBLE OPERAND INSTRUCTIONS.  IT SETS
 7904                                ;DATA IN R0 AND LOCATION 0 AND OPERATES UPON IT USING DOP INSTRUCTIONS.
 7905                                ;
 7906                                ;*****************************************************************************
  (2)                               ;TEST 117       TEST MODE 0,X DOUBLE-OPERAND INSTRUCTIONS
  (3)                               ;*****************************************************************************
  (2)  007114                       TS117:
 7907  007114  005000                        CLR     R0              ;R0=0
 7908  007116  005010                        CLR     (R0)            ;LOC. 0=0
 7909  007120  105110                        COMB    (R0)            ;LOC. 0=377
 7910  007122  005220                        INC     (R0)+           ;LOC. 0=400   R0=2
 7911  007124  005400                        NEG     R0              ;R0=-2
 7912  007126  060037  000000                ADD     R0,@#0          ;TRY ADD 0,3;  LOC. 0=376
 7913  007132  100403                        BMI     DOP03A          ;CC=0001?
 7914  007134  001402                        BEQ     DOP03A
 7915  007136  102401                        BVS     DOP03A
 7916  007140  103401                        BCS     DOP03B
  (1)  007142                       DOP03A:
  (2)  007142  104000                        EMT                     ;CC'S NOT SET CORRECTLY
 7917  007144  105137  000000        DOP03B: COMB    @#0             ;LOC. 0=1
 7918  007150  005337  000000                DEC     @#0             ;LOC. 0=0
 7919  007154  001401                        BEQ     TS120
  (3)  007156  104000                        EMT                     ;DATA RESULT INCORRECT
 7920                                ;*****************************************************************************
 7921                                ;
 7922                                ;       THIS TEST VERIFIES MODE 0,0 DOP NON-MODIFYING INSTRUCTIONS.
 7923                                ;R0 AND R4 ARE PRESET TO 0 AND 1 RESPECTIVELY.  COMPARE INSTRUCTIONS ARE
 7924                                ;THEN EXECUTED AND CHECKED.  FIRST R4 IS COMPARED TO R0 THEN R0 TO R4.
 7925                                ;
 7926                                ;*****************************************************************************
  (2)                               ;TEST 120       TEST DOP NON-MODIFYING INST. W/ SOURCE MODE 0,0
  (3)                               ;*****************************************************************************
  (2)  007160                       TS120:
 7927  007160  005000                        CLR     R0              ;R0=0
 7928  007162  005004                        CLR     R4              ;R4=0
 7929  007164  005204                        INC     R4              ;R4=1
 7930  007166  020400                        CMP     R4,R0           ;TRY COMPARE R4 TO R0
 7931  007170  003001                        BGT     DNM1
  (2)  007172  104000                        EMT                     ;CC'S NOT CORRECT FOR CMP
```

```
 7932  007174  020004                DNM1:   CMP     R0,R4           ;TRY COMPARE R0 TO R4
 7933  007176  002401                        BLT     DNM2
  (2)  007200  104000                        EMT                     ;CC'S NOT CORRECT FOR CMP
 7934  007202  005200                DNM2:   INC     R0              ;R0=1
 7935  007204  020400                        CMP     R4,R0           ;TRY COMPARE R4=1 TO R0=1
 7936  007206  001401                        BEQ     DNM3
  (2)  007210  104000                        EMT                     ;CC'S NOT CORRECT (Z=1) FOR CMP
 7937  007212  005000                DNM3:   CLR     R0              ;R0=0
 7938  007214  005100                        COM     R0              ;R0=177777
 7939  007216  005004                        CLR     R4              ;R4=0
 7940  007220  030004                        BIT     R0,R4           ;TRY BIT R0 TO R4
 7941  007222  001401                        BEQ     DNM4
  (2)  007224  104000                        EMT                     ;CC'S NOT CORRECT FOR BIT
 7942  007226  005304                DNM4:   DEC     R4              ;R4=177777
 7943  007230  030004                        BIT     R0,R4           ;TRY BIT AGAIN
 7944  007232  100401                        BMI     TS121
  (3)  007234  104000                        EMT                     ;CC'S NOT CORRECT FOR BIT
 7945                                ;*******************************************************************************
 7946                                ;
 7947                                ;       THIS TEST VERIFIES MODE 0,X DOUBLE OPERAND NON-MODIFYING INSTRUCTIONS.
 7948                                ;IT SETS DATA IN R0 AND LOCATION 0 AND COMPARES THEM USING DOPNM INSTRUCTIONS.
 7949                                ;
 7950                                ;*******************************************************************************
  (2)                               ;TEST 121       TEST MODE 0,X DOUBLE-OPERAND NON-MODIFYING INSTS.
  (3)                               ;*******************************************************************************
  (2)  007236                       TS121:
 7951  007236  005000                        CLR     R0              ;R0=0
 7952  007240  005010                        CLR     (R0)            ;LOC. 0=0
 7953  007242  005110                        COM     (R0)            ;LOC. 0=177777
 7954  007244  005200                        INC     R0              ;R0=1
 7955  007246  020037  000000                CMP     R0,@#0          ;TRY CMP MODE 0,3
 7956  007252  100403                        BMI     DNM103A         ;CC=0001
 7957  007254  001402                        BEQ     DNM103A
 7958  007256  102401                        BVS     DNM103A
 7959  007260  103401                        BCS     DNM103B
  (1)  007262                       DNM103A:
  (2)  007262  104000                        EMT                     ;CC'S NOT SET CORRECTLY
 7960  007264  005300                DNM103B: DEC    R0
 7961  007266  001002                        BNE     DNM103C
 7962  007270  005210                        INC     (R0)
 7963  007272  001401                        BEQ     TS122
  (2)  007274                       DNM103C:
  (3)  007274  104000                        EMT                     ;DATA INCORRECTLY MODIFIED BY CMP
 7964                                ;*******************************************************************************
 7965                                ;
 7966                                ;       THIS TEST VERIFIES MODE 1 DOP INSTRUCTIONS.  R0 IS SET TO -1
 7967                                ;AND LOC 0 TO 1.  R4 IS THEN CLEARED AND USED TO POINT TO LOC 0.
 7968                                ;IN THE ADD MODE 1 INSTRUCTION, LOC 0 IS ADDED TO R0 AND THE
 7969                                ;RESULTS VERIFIED.
 7970                                ;
 7971                                ;*******************************************************************************
  (2)                               ;TEST 122       TEST MODE 1 W/ DOP INST.
  (3)                               ;*******************************************************************************
  (2)  007276                       TS122:
 7972  007276  005000                        CLR     R0              ;R0=0
 7973  007300  005100                        COM     R0              ;R0=177777
```

```
 7974  007302  005004                    CLR     R4              ;R4=0
 7975  007304  005014                    CLR     (R4)            ;LOC 0=0
 7976  007306  005214                    INC     (R4)            ;LOC 0=1
 7977  007310  061400                    ADD     (R4),R0         ;TRY ADD SOURCE MODE 1
 7978  007312  001401                    BEQ     TS123
  (3)  007314  104000                    EMT                     ;RESULT OF ADD INCORRECT
 7979
 7980                            ;******************************************************************************
 7981                            ;
 7982                            ;       THIS TEST VERIFIES MODE 1 DOP BYTE INSTRUCTIONS WHICH ADDRESS
 7983                            ;EVEN BYTES.  LOC. 0 IS SET TO -1 AND R4 IS CLEARED.  THEN R4 IS
 7984                            ;SET TO -1 USING A BISB THRU R0 WITH MODE 1.
 7985                            ;
 7986                            ;******************************************************************************
  (2)                            ;TEST 123       TEST MODE 1 - EVEN BYTE W/ DOP INSTS.
  (3)                            ;******************************************************************************
  (2)  007316                   TS123:
 7987  007316  005000                    CLR     R0              ;R0=0
 7988  007320  005010                    CLR     (R0)            ;LOC. 0=0
 7989  007322  005110                    COM     (R0)            ;LOC. 0=177777
 7990  007324  005004                    CLR     R4              ;R4=0
 7991  007326  151004                    BISB    (R0),R4         ;TRY MODE 1- EVEN BYTE W/ DOP
 7992  007330  105104                    COMB    R4              ;R4=0
 7993  007332  001401                    BEQ     TS124
  (3)  007334  104000                    EMT                     ;RESULT OF BISB IS INCORRECT
 7994
 7995                            ;******************************************************************************
 7996                            ;
 7997                            ;       THIS TEST VERIFIES MODE 1 DOP NON-MODIFYING INSTRUCTIONS
 7998                            ;WHICH ADDRESS EVEN BYTES.  LOC. 0 IS SET TO -1 AND R0 IS CLEARED
 7999                            ;AND USED AS THE ADDRESSING REGISTER.  R4 IS SET TO 377 AND A
 8000                            ;MODE 1,0 CMPB INSTRUCTION IS USED THE RESULTS VERIFIED.
 8001                            ;
 8002                            ;******************************************************************************
  (2)                            ;TEST 124       TEST MODE 1 - EVEN BYTE W/ DOP NON-MODIFYING INST.
  (3)                            ;******************************************************************************
  (2)  007336                   TS124:
 8003  007336  005000                    CLR     R0              ;R0=0
 8004  007340  005010                    CLR     (R0)            ;LOC 0=0
 8005  007342  005110                    COM     (R0)            ;LOC 0=177777
 8006  007344  005004                    CLR     R4              ;R4=0
 8007  007346  105104                    COMB    R4              ;R4=377
 8008  007350  121004                    CMPB    (R0),R4         ;TRY MODE 1 - EVEN BYTE W/ DOP NON-MODIFYING
 8009  007352  001401                    BEQ     TS125
  (3)  007354  104000                    EMT                     ;RESULT OF CMPB INCORRECT
 8010
 8011                            ;******************************************************************************
 8012                            ;
 8013                            ;       THIS TEST VERIFIES MODE 1,0 MOVB INSTRUCTIONS
 8014                            ;WHICH ADDRESS EVEN BYTES.  LOC. 0 IS SET TO 177400, R0 IS CLEARED AND
 8015                            ;R4 IS SET TO -1.  MOVB ARE USED TO MOVE BYTE 0 TO R4.  THIS
 8016                            ;VERIFIES THAT THE PROPER BYTE WAS SELECTED AND THAT THE SIGN-X-TEND
 8017                            ;FUNCTION WITH MODE 0.
 8018                            ;       THEN LOC. 0 IS COMPLEMENTED AND THE SAME PROCEDURE EXERCISES
 8019                            ;THE LOGIC FOR COMPLEMENTARY DATA.
 8020                            ;       THIS TEST EXERCISES UNIQUE MICROCODE.
```

```
                                ;
                                ;*******************************************************************************
                                ;TEST 125        TEST MOV INSTRUCTION MODE 1,0 EVEN BYTE
                                ;*******************************************************************************
007356                          TS125:
007356  005000                          CLR     R0              ;R0=0
007360  005010                          CLR     (R0)            ;LOC 0=0
007362  105110                          COMB    (R0)            ;LOC 0=177400
007364  005110                          COM     (R0)
007366  005004                          CLR     R4              ;R4=0
007370  005104                          COM     R4              ;R4=177777
007372  111004                          MOVB    (R0),R4         ;R4=0
007374  005704                          TST     R4              ;CHECK SIGN OF WORD
007376  001401                          BEQ     DOP1
007400  104000                          EMT                     ;MOVB SHOULD SIGN X-TEND
007402  005110                  DOP1:   COM     (R0)            ;LOC 0=177777
007404  111004                          MOVB    (R0),R4         ;DO MOVB W/ EVEN BYTE
007406  100401                          BMI     TS126
007410  104000                          EMT                     ;MOVB SHOULD SIGN X-TEND

                                ;*******************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 1 DOP INSTRUCTIONS WHICH REFERENCE
                                ;ODD BYTES.  LOC. 0 IS SET TO 177400.  R0 IS SET TO 0 AND R4 IS
                                ;SET TO 1.  THE BISB INSTRUCTION USES THE DATA IN BYTE 1 TO SET BYTE 0.
                                ;THE RESULT IS CHECKED BY INCREMENTING THE WORD (LOC. 0) TO ZERO.
                                ;
                                ;*******************************************************************************
                                ;TEST 126        TEST MODE 1-ODD BYTE W/ DOP INSTS.
                                ;*******************************************************************************
007412                          TS126:
007412  005000                          CLR     R0              ;R0=0
007414  005010                          CLR     (R0)            ;LOC. 0=0
007416  005004                          CLR     R4              ;R4=0
007420  005204                          INC     R4              ;R4=1
007422  105114                          COMB    (R4)            ;LOC. 0=177400
007424  151410                          BISB    (R4),(R0)       ;TRY TO BIS LOW ORDER BITS W/ MODE 1
007426  005210                          INC     (R0)            ;CHECK RESULT
007430  001401                          BEQ     TS127
007432  104000                          EMT                     ;RESULT OF BISB INCORRECT

                                ;*******************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 2 DOP INSTRUCTIONS.  LOC. 0 IS SET TO -1.
                                ;R0 IS CLEARED AND USED AS THE MODE 2 ADDRESSING REGISTER TO MOVE LOC. 0
                                ;TO R7.  THE DATA RESULTS ARE VERIFIED AND THE INCREMENTING OF THE REGISTER
                                ;IS CHECKED.
                                ;
                                ;*******************************************************************************
                                ;TEST 127        TEST MODE 2 W/ DOP INSTS.
                                ;*******************************************************************************
007434                          TS127:
007434  005000                          CLR     R0              ;R0=0
007436  005010                          CLR     (R0)            ;LOC. 0=0
007440  005110                          COM     (R0)            ;LOC. 0=177777
007442  012004                          MOV     (R0)+,R4        ;TRY MOVE MODE 2,0
```

F5



```
007444  005204                          INC     R4              ;CHECK R4
007446  001401                          BEQ     DOP2
007450  104000                          EMT                     ;RESULT OF MOV INST INCORRECT
007452  005300                  DOP2:   DEC     R0              ;TEST R0 AFTER MODE 2
007454  005300                          DEC     R0
007456  001401                          BEQ     TS130
007460  104000                          EMT                     ;REGISTER NOT INCREMENTED IN MODE 2

                                ;*******************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 2 DOP BYTE INSTRUCTIONS WHICH ADDRESS
                                ;EVEN BYTES.  LOC. 0 IS SET TO -1.  R0 IS CLEARED AND USED AS THE
                                ;ADDRESSING REGISTER IN A TEST WHICH TRIES TO CLEAR BYTE 1 USING
                                ;BYTE 0 DATA AND A BICB.  UNIQUE IN THIS TEST IS USE OF THE
                                ;SAME ADDRESSING REGISTER FOR BOTH SOURCE AND DESTINATION.  THE SOURCE AND
                                ;DESTINATION IS CHECKED TO INSURE PROPER FUNCTIONING.
                                ;
                                ;*******************************************************************************
                                ;TEST 130       TEST MODE 2 - EVEN BYTE W/ DOP INST.
                                ;*******************************************************************************
007462                          TS130:
007462  005000                          CLR     R0              ;R0=0
007464  010010                          MOV     R0,(R0)         ;LOC. 0=0
007466  005110                          COM     (R0)            ;LOC. 0=177777
007470  142010                          BICB    (R0)+,(R0)      ;TRY TO CLEAR BYTE 1 FROM BYTE 0 W/ BICB
007472  105737  000001                  TSTB    @#1             ;CHECK RESULT
007476  001401                          BEQ     DOPB2A
007500  104000                          EMT                     ;BICB DESTINATION INCORRECT
007502  105137  000000          DOPB2A: COMB    @#0             ;CHECK BICB SOURCE
007506  001401                          BEQ     TS131
007510  104000                          EMT                     ;BICB SOURCE INCORRECTLY CHANGED
                                ;*******************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 2 DOP BYTE INSTRUCTIONS WHICH REFERENCE
                                ;ODD BYTES.  R0 IS SET TO 1, LOC. 0 IS SET TO 177400, AND R4 IS CLEARED.
                                ;A MODE 2 MOVB USES R0 TO MOVE BYTE 1 TO R4.  AN INCREMENT
                                ;IS USED TO CHECK THAT THE PROPER BYTE WAS MOVED AND SIGN X-TENDED.
                                ;
                                ;*******************************************************************************
                                ;TEST 131       TEST MODE 2 - ODD BYTE W/ DOP INST.
                                ;*******************************************************************************
007512                          TS131:
007512  005000                          CLR     R0              ;R0=0
007514  005004                          CLR     R4              ;R4=0
007516  005010                          CLR     (R0)            ;LOC. 0=0
007520  005110                          COM     (R0)            ;LOC. 0=177777
007522  105120                          COMB    (R0)+           ;LOC 0=177400; R0=1
007524  112004                          MOVB    (R0)+,R4        ;TRY DOP MODE 2 W/ ODD BYTE
007526  005204                          INC     R4              ;CHECK RESULT OF MOVB
007530  001401                          BEQ     DOPB2B
007532  104000                          EMT                     ;RESULT OF MOVB INCORRECT
007534  005740                  DOPB2B: TST     -(R0)           ;BUMP R0 DOWN BY 2
007536  005700                          TST     R0              ;CHECK R0
007540  001401                          BEQ     TS132
007542  104000                          EMT                     ;MODE 2 BYTE DID NOT INCREMENT REG. CORRECTLY
                                ;*******************************************************************************
```

```
                                        ;
                                        ;       THIS TEST VERIFIES MODE 3 DOUBLE-OPERAND INSTRUCTIONS.
                                        ;LOC. 0 IS LOADED WITH ALTERNATING ZEROES AND ONES; AND R0 IS LOADED
                                        ;WITH ALTERNATING ONES AND ZEROES.  A MODE 3 BIS IS USED TO SET R0
                                        ;TO -1 BY USING LOC. 0 AS THE SOURCE TO BIS THE ZEROES IN R0.  THE
                                        ;RESULT IS TESTED BY INCREMENTING R0 AND CHECKING FOR ZERO.
                                        ;
                                        ;*****************************************************************************
                                        ;TEST 132       TEST MODE 3 W/ DOP INSTS.
                                        ;*****************************************************************************
 (2)  007544                            TS132:
8117  007544  012737  052525  000000          MOV     #052525,@#0     ;MOVE 52525 TO LOC. 0
8118  007552  012700  125252                  MOV     #125252,R0      ;SET ALT. ONE AND ZERO IN R0
8119  007556  053700  000000                  BIS     @#0,R0          ;TRY TO SET ALL OTHER BITS W/ MODE 3
8120  007562  005200                          INC     R0              ;TEST RESULT
8121  007564  001401                          BEQ     TS133
 (3)  007566  104000                          EMT                     ;BIS W/ MODE 3 INCORRECT RESULT
                                        ;*****************************************************************************
                                        ;
                                        ;       THIS TEST VERIFIES MODE 3 DOUBLE OPERAND BYTE INSTRUCTIONS WHICH
                                        ;ADDRESS EVEN BYTES.  BYTE 0 IS SET TO ALTERNATING 1'S AND 0'S; BYTE 1,
                                        ;ALTERNATING 0'S AND 1'S.  R0 IS CLEARED AND A BISB IS USED TO
                                        ;SET THE LOW BYTE OF R0 TO 252.
                                        ;
                                        ;*****************************************************************************
                                        ;TEST 133       TEST MODE 3 - EVEN BYTE W/ DOP INSTS.
                                        ;*****************************************************************************
 (2)  007570                            TS133:
8130  007570  012737  052652  000000          MOV     #52652,@#0      ;MOVE 1'S AND 0' PATTERN TO LOC. 0
8131  007576  005000                          CLR     R0              ;R0=0
8132  007600  153700  000000                  BISB    @#0,R0          ;TRY R0=252 W/ MODE 3 - EVEN BYTE
8133  007604  022700  000252                  CMP     #252,R0         ;BISB W/ EVEN BYTE SUCCESSFUL?
8134  007610  001401                          BEQ     TS134
 (3)  007612  104000                          EMT                     ;BISB W/ MODE 3 - EVEN BYTE FAILED
                                        ;*****************************************************************************
                                        ;
                                        ;       THIS TEST VERIFIES MODE 3 DOUBLE OPERAND BYTE INSTRUCTIONS
                                        ;WHICH ADDRESS ODD BYTES.  THE SAME PROCEDURE USED IN PREVIOUS
                                        ;TEST IS USED HERE.  THIS TIME BYTE 1 IS USED AS THE SOURCE BYTE.
                                        ;THE EXPECTED RESULT IS:  R0 = 125.
                                        ;
                                        ;*****************************************************************************
                                        ;TEST 134       TEST MODE 3 - ODD BYTE W/ DOP INSTS.
                                        ;*****************************************************************************
 (2)  007614                            TS134:
8143  007614  012737  052652  000000          MOV     #52652,@#0      ;MOVE 1'S AND 0'S PATTERN TO LOC 0
8144  007622  005000                          CLR     R0              ;R0=0
8145  007624  153700  000001                  BISB    @#1,R0          ;TRY R0=152 W/ MODE 3 - ODD BYTE
8146  007630  022700  000125                  CMP     #125,R0         ;R0=125?
8147  007634  001401                          BEQ     TS135
 (3)  007636  104000                          EMT                     ;BISB W/ MODE 3 - ODD BYTE FAILED

                                        ;*****************************************************************************
                                        ;TEST 135       TEST DEST. MODE 0-BYTE W/ DOP NON-MODIFYING MST
                                        ;*****************************************************************************
 (2)  007640                            TS135:
```

```
8150  007640  005000                        CLR     R0              ;R0=0
8151  007642  105100                        COMB    R0              ;R0=377
8152  007644  000263                        *SEC!SEV                ;SET C AND V BITS
8153  007646  132700  000200                BITB    #200,R0         ;TRY DOPNM DEST. MODE 0-BYTE
8154  007652  001403                        BEQ     DNMB0A          ;BR TO ERROR IF Z BIT SET
8155  007654  102402                        BVS     DNMB0A          ;BR TO ERROR IF V BIT SET
8156  007656  103001                        BCC     DNMB0A          ;BR TO ERROR IF C BIT CLEAR.
8157  007660  100401                        BMI     DNMB0B
 (1)  007662                          DNMB0A:
 (2)  007662  104000                        EMT                     ;CC'S INCORRECT
 (2)  007664  105100                  DNMB0B: COMB    R0              ;CHECK DESTINATION DATA
8159  007666  001401                        BEQ     TS136
 (3)  007670  104000                        EMT                     ;DEST. DATA MODIFIED
8160
8161                                  ;*****************************************************************************
 (2)                                  ;TEST 136       TEST DEST. MODE 1 W/ DOP NON-MODIFYING INST
 (3)                                  ;*****************************************************************************
 (2)  007672                          TS136:
8162  007672  005000                        CLR     R0              ;R0=0
8163  007674  005010                        CLR     (R0)            ;LOC. 0=0
8164  007676  000241                        CLC                     ;CLEAR C BIT
8165  007700  032710  177777                BIT     #177777,(R0)    ;TRY DOPNM DEST. MODE 1
8166  007704  100403                        BMI     DNM1A           ;BR TO ERROR IF N BIT SET
8167  007706  102402                        BVS     DNM1A           ;BR TO ERROR IF V BIT SET
8168  007710  103401                        BCS     DNM1A           ;BR TO ERROR IF C BIT SET
8169  007712  001401                        BEQ     DNM1B
 (1)  007714                          DNM1A:
 (2)  007714  104000                        EMT                     ;COND. CODES INCORRECT
8170  007716  005710                  DNM1B: TST     (R0)            ;CHECK TEST DATA
8171  007720  001401                        BEQ     TS137
 (3)  007722  104000                        EMT                     ;DESTINATION DATA MODIFIED
8172
8173                                  ;*****************************************************************************
 (2)                                  ;TEST 137       TEST DEST, MODE 2 W/ DOP NON-MODIFYING INST.
 (3)                                  ;*****************************************************************************
 (2)  007724                          TS137:
8174  007724  005000                        CLR     R0              ;R0=0
8175  007726  005010                        CLR     (R0)            ;LOC. 0=0
8176  007730  052710  125252                BIS     #125252,(R0)    ;LOC. 0=125252
8177  007734  032720  077777                BIT     #77777,(R0)+    ;TRY DOPNM INST W/ MODE 2
8178  007740  102402                        BVS     DNM2A           ;BR TO ERROR IF V BIT SET
8179  007742  001401                        BEQ     DNM2A           ;BR TO ERROR IF Z-BIT SET
8180  007744  100001                        BPL     DNM2B
 (1)  007746                          DNM2A:
 (2)  007746  104000                        EMT                     ;COND. CODES INCORRECT
8181  007750  005300                  DNM2B: DEC     R0              ;DECREMENT R0 TO CHECK IT.
8182  007752  005300                        DEC     R0
8183  007754  001401                        BEQ     DNM2D
 (1)  007756                          DNM2C:
 (2)  007756  104000                        EMT                     ;MODE 2 REGISTER NOT INCREMENTED BY 2
8184  007760  022710  125252          DNM2D: CMP     #125252,(R0)    ;CHECK DEST. DATA
8185  007764  001401                        BEQ     TS140
 (3)  007766  104000                        EMT                     ;DEST. DATA MODIFIED
8186
8187                                  ;*****************************************************************************
 (2)                                  ;TEST 140       TEST DEST. MODE 2-BYTE, W/DOP NON-MODIFYING INST
```

```
 (3)                                ;*****************************************************************************
 (2)  007770                        TS140:
8188  007770  005000                        CLR     R0              ;R0=0
8189  007772  005010                        CLR     (R0)            ;LOC. 0=0
8190  007774  052710  052652                BIS     #52652,(R0)     ;LOC. 0=52652
8191  010000  000263                        +SEC!SEV                ;SET C AND V BITS
8192  010002  132720  000201                BITB    #201,(R0)+      ;TRY DOPNM INST. W/ MODE 2 EVEN BYTE
8193  010006  001403                        BEQ     DNMB2A          ;BR TO ERROR IF Z-BIT SET
8194  010010  103002                        BCC     DNMB2A          ;BR TO ERROR IF C-BIT CLEAR
8195  010012  102401                        BVS     DNMB2A          ;BR TO ERROR IF V-BIT SET
8196  010014  100401                        BMI     DNMB2B
 (1)  010016                        DNMB2A:
 (2)  010016  104000                        EMT                     ;COND. CODES INCORRECT
8197  010020  005300                DNMB2B: DEC     R0              ;CHECK DEST. REGISTER.
8198  010022  001401                        BEQ     DNMB2C
 (2)  010024  104000                        EMT                     ;DEST. REGISTER NOT INCREMENTED BY 1
8199  010026  005200                DNMB2C: INC     R0              ;R0=1
8200  010030  132720  000201                BITB    #201,(R0)+      ;TRY DOPNM INST. W/MODE 2-ODD BYTE
8201  010034  001402                        BEQ     DNMB2D  ;BR TO ERROR IF Z-BIT SET
8202  010036  102401                        BVS     DNMB2D          ;BR TO ERROR IF V-BIT SET
8203  010040  100001                        BPL     DNMB2E
 (1)  010042                        DNMB2D:
 (2)  010042  104000                        EMT                     ;COND. CODES INCORRECT
8204  010044  005300                DNMB2E: DEC     R0              ;DEC R0 TO CHECK IT.
8205  010046  005300                        DEC     R0
8206  010050  001401                        BEQ     DNMB2F
 (2)  010052  104000                        EMT                     ;DEST. REGISTER NOT INCREMENTED BY 1
8207  010054  022710  052652        DNMB2F: CMP     #52652,(R0)     ;CHECK DEST. DATA IS UNMODIFIED
8208  010060  001401                        BEQ     TS141
 (3)  010062  104000                        EMT                     ;DEST. DATA WAS MODIFIED.
8209
8210
8211                                ;*****************************************************************************
 (2)                                ;TEST 141       TEST DEST. MODE 3-BYTES W/DOP NON-MODIFYING INST.
 (3)                                ;*****************************************************************************
 (2)  010064                        TS141:
8212  010064  005000                        CLR     R0              ;R0=0
8213  010066  005010                        CLR     (R0)            ;LOC. 0=0
8214  010070  052710  125125                BIS     #125125,(R0)    ;LOC. 0=125125
8215  010074  105100                        COMB    R0              ;R0=377
8216  010076  005200                        INC     R0              ;R0=400
8217  010100  005010                        CLR     (R0)            ;LOC. 400=0
8218  010102  000263                        +SEC!SEV                ;C-BIT=V-BIT=1
8219  010104  132730  000201                BITB    #201,@(R0)+     ;TRY DOPNM W/MODE 3-EVEN BYTE
8220  010110  001403                        BEQ     DNMB3A          ;BR TO ERROR IF Z BIT SET
8221  010112  102402                        BVS     DNMB3A          ;BR TO ERROR IF V BIT SET
8222  010114  103001                        BCC     DNMB3A          ;BR TO ERROR IF C BIT CLEAR
8223  010116  100001                        BPL     DNMB3B
 (1)  010120                        DNMB3A:
 (2)  010120  104000                        EMT                     ;COND. CODES INCORRECT
8224  010122  022700  000402        DNMB3B: CMP     #402,R0         ;CHECK DEST. REGISTER INC. BY 2 AND INC BY 2 AGAIN
8225  010126  001401                        BEQ     DNMB3C
 (2)  010130  104000                        EMT                     ;DEST. REGISTER NOT INCREMENTED BY 2
8226  010132  005200                DNMB3C: INC     R0              ;R0=404
8227  010134  005200                        INC     R0
8228  010136  132730  000201                BITB    #201,@(R0)+     ;TRY DOPNM DEST MODE 3-BYTE(ODD)
```

```
 8229  010142  001402                        BEQ     DNMB3D          ;BR TO ERROR IF Z BIT SET
 8230  010144  102401                        BVS     DNMB3D          ;BR TO ERROR IF V BIT SET
 8231  010146  100401                        BMI     DNMB3E
  (1)  010150                          DNMB3D:
  (2)  010150  104000                        EMT                     ;COND. CODES INCORRECT
 8232  010152  005004                  DNMB3E: CLR     R4              ;R4=0
 8233  010154  022714  125125                CMP     #125125,(R4)    ;CHECK DEST. DATA
 8234  010160  001401                        BEQ     TS142
  (3)  010162  104000                        EMT                     ;DEST. DATA MODIFIED
 8235
 8236                                  ;*************************************************************************************
  (2)                                  ;TEST 142       TEST DEST. MODE 4 W/DOP NON-MODIFYING INST.
  (3)                                  ;*************************************************************************************
  (2)  010164                          TS142:
 8237  010164  005000                        CLR     R0              ;R0=0
 8238  010166  005010                        CLR     (R0)            ;LOC. 0=0
 8239  010170  052710  125252                BIS     #125252,(R0)    ;LOC. 0=125125
 8240  010174  052700  000002                BIS     #2,R0           ;R0=2
 8241  010200  000277                        SCC                     ;SET ALL COND. CODE BITS
 8242  010202  032740  020000                BIT     #20000,-(R0)    ;TRY DOPNM W/ MODE 4
 8243  010206  100403                        BMI     DNM4A           ;BR TO ERROR IF N-BIT SET
 8244  010210  102402                        BVS     DNM4A           ;BR TO ERROR IF V-BIT SET
 8245  010212  103001                        BCC     DNM4A           ;BR TO ERROR IF C-BIT CHAR
 8246  010214  001001                        BNE     DNM4B
  (1)  010216                          DNM4A:
  (2)  010216  104000                        EMT                     ;COND. CODES INCORRECT
 8247  010220  005700                  DNM4B: TST     R0              ;CHECK DEST. REGISTER
 8248  010222  001401                        BEQ     DNM4C
  (2)  010224  104000                        EMT                     ;DEST. REGISTER NOT DECREMENTED BY 2
 8249  010226  022737  125252  000000  DNM4C: CMP     #125252,@#0     ;CHECK DEST. DATA
 8250  010234  001401                        BEQ     TS143
  (3)  010236  104000                        EMT                     ;DEST. DATA MODIFIED
 8251
 8252                                  ;*************************************************************************************
  (2)                                  ;TEST 143       TEST DEST. MODE 4-BYTE W/ DOP NON-MODIFYING INST.
  (3)                                  ;*************************************************************************************
  (2)  010240                          TS143:
 8253  010240  005000                        CLR     R0              ;R0=0
 8254  010242  005010                        CLR     (R0)            ;LOC. 0=0
 8255  010244  052710  052652                BIS     #52652,(R0)     ;LOC. 0=52652
 8256  010250  052700  000002                BIS     #2,R0           ;R0=2
 8257  010254  000257                        CCC                     ;COND. CODES=0
 8258  010256  132740  000201                BITB    #201,-(R0)      ;TRY DOPNM INST W/MODE 4 ODD BYTE
 8259  010262  102403                        BVS     DNMB4A          ;BR TO ERROR IF V BIT SET
 8260  010264  001402                        BEQ     DNMB4A  ;BR TO ERROR IF Z BIT SET
 8261  010266  103401                        BCS     DNMB4A          ;BR TO ERROR IF C BIT SET
 8262  010270  001001                        BNE     DNMB4B
  (1)  010272                          DNMB4A:
  (2)  010272  104000                        EMT                     ;COND. CODES INCORRECT
 8263  010274  022700  000001          DNMB4B: CMP     #1,R0           ;CHECK DEST. REGISTER
 8264  010300  001401                        BEQ     DNMB4C
  (2)  010302  104000                        EMT                     ;DEST REG. NOT DECREMENTED BY 1
 8265  010304  132740  000201          DNMB4C: BITB    #201,-(R0)      ;TRY DOPNM INST. W/MODE 4 EVEN BYTE
 8266  010310  001401                        BEQ     DNMB4D          ;BR TO ERROR IF Z-BIT SET
 8267  010312  100401                        BMI     DNMB4E
  (1)  010314                          DNMB4D:
```

```
 (2)  010314  104000                          EMT                     ;COND. CODES INCORRECT
8268  010316  005700                DNMB4E:   TST     R0              ;CHECK DEST. REGISTER
8269  010320  001401                          BEQ     DNMB4F
 (2)  010322  104000                          EMT                     ;DEST. REG. NOT DECREMENTED BY 1
8270  010324  022710  052652        DNMB4F:   CMP     #52652,(R0)     ;CHECK DESTINATION DATA
8271  010330  001401                          BEQ     TS144
 (3)  010332  104000                          EMT                     ;DEST. DATA MODIFIED
8272
8273                                ;*************************************************************************
 (2)                                ;TEST 144       TEST DEST MODE 5 W/DOP NON-MODIFYING INST.
 (3)                                ;*************************************************************************
 (2)  010334                        TS144:
8274  010334  005000                          CLR     R0              ;R0=0
8275  010336  005010                          CLR     (R0)            ;LOC 0=0
8276  010340  052710  100000                  BIS     #100000,(R0)    ;LOC. 0=100000
8277  010344  052700  000402                  BIS     #402,R0         ;R0=2
8278  010350  000277                          SCC                     ;SET ALL COND. CODE BITS
8279  010352  032750  100000                  BIT     #100000,@-(R0)  ;TRY DOPNM W/MODE 5
8280  010356  102403                          BVS     DNM5A           ;BR TO ERROR IF V-BIT SET
8281  010360  103002                          BCC     DNM5A           ;BR TO ERROR IF C-BIT CLEAR
8282  010362  001401                          BEQ     DNM5A           ;BR TO ERROR IF Z-BIT SET
8283  010364  100401                          BMI     DNM5B
 (1)  010366                        DNM5A:
 (2)  010366  104000                          EMT                     ;COND. CODES INCORRECT
8284  010370  022700  000400        DNM5B:    CMP #400,R0             ;CHECK DEST. REGISTER
8285  010374  001401                          BEQ     DNM5C
 (2)  010376  104000                          EMT                     ;DEST. REGISTER NOT DECREMENTED BY 2
8286  010400  022737  100000  000000  DNM5C:  CMP     #100000,@#0     ;CHECK DESTINATION DATA
8287  010406  001401                          BEQ     TS145
 (3)  010410  104000                          EMT                     ;DEST. DATA INCORRECTLY MODIFIED
8288
8289                                ;*************************************************************************
 (2)                                ;TEST 145       TEST DEST. MODE 6 W/DOP NON-MODIFYING INST.
 (3)                                ;*************************************************************************
 (2)  010412                        TS145:
8290  010412  005000                          CLR     R0              ;R0=0
8291  010414  005010                          CLR     (R0)            ;LOC> 0=0
8292  010416  052710  000001                  BIS     #1,(R0)         ;LOC. 0=1
8293  010422  005100                          COM     R0              ;R0=-1 C-BIT=1
8294  010424  032760  000001  000001          BIT     #1,1(R0)        ;TRY DOPNM W/MODE 6
8295  010432  001403                          BEQ     DNM6A           ;BR TO ERROR IF Z-BIT SET
8296  010434  102402                          BVS     DNM6A           ;BR TO ERROR IF V-BIT SET
8297  010436  103001                          BCC     DNM6A           ;BR TO ERROR IF C-BIT CLEAR
8298  010440  100001                          BPL     DNM6B
 (1)  010442                        DNM6A:
 (2)  010442  104000                          EMT                     ;COND CODES INCORRECT
8299  010444  022700  177777        DNM6B:    CMP     #-1,R0          ;CHECK DEST. REGISTER
8300  010450  001401                          BEQ     DNM6C
 (2)  010452  104000                          EMT                     ;DEST. REGISTER MODIFIED
8301  010454  022737  000001  000000  DNM6C:  CMP     #1,@#0          ;CHECK DEST. DATA
8302  010462  001401                          BEQ     TS146
 (3)  010464  104000                          EMT                     ;DEST. DATA MODIFIED
8303
8304                                ;*************************************************************************
 (2)                                ;TEST 146       TEST DEST MODE 7 W/DOP NON-MODIFYING INST.
 (3)                                ;*************************************************************************
```

```
  (2)  010466                          TS146:
8305   010466  005000                         CLR     R0              ;R0=0
8306   010470  005010                         CLR     (R0)            ;LOC. 0=0 C-BIT=0
8307   010472  052710  125125                 BIS     #125125,(R0)    ;LOC. 0=125125
8308   010476  052700  000001                 BIS     #1,R0           ;R0=1
8309   010502  132770  000125  000403         BITB    #125,@403(R0)   ;TRY DOPNM W/MODE 7
8310   010510  102403                         BVS     DNM7A           ;BR TO ERROR IF V-BIT SET
8311   010512  100402                         BMI     DNM7A           ;BR TO ERROR IF N-BIT SET
8312   010514  103401                         BCS     DNM7A           ;BR TO ERROR IF C-BIT SET
8313   010516  001401                         BEQ     DNM7B
  (1)  010520                          DNM7A:
  (2)  010520  104000                         EMT                     ;COND. CODES INCORRECT
8314   010522  022700  000001         DNM7B:  CMP     #1,R0           ;CHECK DEST. REGISTER
8315   010526  001401                         BEQ     DNM7C
  (2)  010530  104000                         EMT                     ;DESTINATION REGISTER MODIFIED
8316   010532  022737  125125  000000 DNM7C:  CMP     #125125,@#0     ;CHECK DEST. DATA
8317   010540  001401                         BEQ     TS147
  (3)  010542  104000                         EMT                     ;DEST. DATA INCORRECT
8318
8319                                 ;*******************************************************************************
8320                                 ;
8321                                 ;       THIS TEST VERIFIES THE MOV DESTINATION MODE 1 INSTRUCTION.
8322                                 ;DATA IS SET IN R0 USING SOP INSTRUCTIONS AND THEN MOVED TO LOC. 0
8323                                 ;USING MOV SRC MODE 0, DEST. MODE 1.
8324                                 ;
8325                                 ;*******************************************************************************
  (2)                                ;TEST 147       TEST MOV DESTINATION MODE 1
  (3)                                ;*******************************************************************************
  (2)  010544                        TS147:
8326   010544  005000                         CLR     R0              ;R0=0
8327   010546  005010                         CLR     (R0)            ;LOC. 0=0
8328   010550  005100                         COM     R0              ;R0=-1
8329   010552  005004                         CLR     R4              ;R4 POINTS TO LOC. 0
8330   010554  010014                         MOV     R0,(R4)         ;TRY MOVE MODE 0,1
8331   010556  102402                         BVS     MDM1A           ;BR TO ERROR IF V SET
8332   010560  001401                         BEQ     MDM1A           ;BR TO ERROR IF Z SET
8333   010562  100401                         BMI     MDM1B
  (1)  010564                        MDM1A:
  (2)  010564  104000                         EMT                     ;CONDITION CODE NOT CORRECT
8334   010566  005704                 MDM1B:  TST     R4
8335   010570  001401                         BEQ     TS150
  (3)  010572  104000                         EMT                     ;DESTINATION REGISTER INCORRECTLY ALTERED
8336
8337                                 ;*******************************************************************************
8338                                 ;
8339                                 ;       THIS TEST VERIFIES THE MOV DESTINATION MODE 2 INSTRUCTION.
8340                                 ;       DATA IS SET IN R0 USING SOP INSTRUCTIONS AND THEN MOVED
8341                                 ;       TO LOCATION 0 USING MOV SRC MODE 0, DEST. MODE 1.
8342                                 ;
8343                                 ;*******************************************************************************
  (2)                                ;TEST 150       TEST MOV DESTINATION MODE 2
  (3)                                ;*******************************************************************************
  (2)  010574                        TS150:
8344   010574  005000                         CLR     R0              ;R0=0
8345   010576  005001                         CLR     R1              ;R1=0
8346   010600  005010                         CLR     (R0)            ;LOC.0=0
```

```
 8347  010602  005110                        COM     (R0)            ;LOC. 0=-1
 8348  010604  010120                        MOV     R1,(R0)+        ;TRY MOVE MODE 0,2
 8349  010606  100402                        BMI     MDM2A           ;BR TO ERROR IF N SET
 8350  010610  102401                        BVS     MDM2A           ;BR TO ERROR IF V SET
 8351  010612  001401                        BEQ     MDM2B
  (1)  010614                          MDM2A:
  (2)  010614  104000                        EMT                     ;CC'S INCORRECT
 8352  010616  005300                  MDM2B: DEC     R0
 8353  010620  005300                        DEC     R0
 8354  010622  001401                        BEQ     MDM2D
  (1)  010624                          MDM2C:
  (2)  010624  104000                        EMT                     ;DESTINATION REGISTER NOT INCREMENTED PROPERLY
 8355  010626  005737  000000          MDM2D: TST     @#0
 8356  010632  001401                        BEQ     TS151
  (3)  010634  104000                        EMT                     ;DESTINATION DATA INCORRECT
 8357
 8358                                  ;*****************************************************************************
 8359                                  ;
 8360                                  ;       THIS TEST VERIFIES DESTINATION MODE 2 W/MOVB INSTS. TWO DIFFERENT MOVB
 8361                                  ;INSTRUCTIONS ARE USED TO MOVE A TEST PATTERN FIRST TO BYTE 0 THEN TO BYTE 1.
 8362                                  ;
 8363                                  ;*****************************************************************************
  (2)                                 ;TEST 151       TEST MOV-BYTE DESTINATION MODE 2
  (3)                                 ;*****************************************************************************
  (2)  010636                         TS151:
 8364  010636  005000                        CLR     R0              ;R0=0
 8365  010640  005010                        CLR     (R0)            ;LOC. 0=0
 8366  010642  112720  000125                MOVB    #125,(R0)+      ;TRY DESTINATION MODE 2 W/EVEN BYTE
 8367  010646  102402                        BVS     MBDM2A          ;BR TO ERROR IF V SET
 8368  010650  001401                        BEQ     MBDM2A          ;BR TO ERROR IF Z SET
 8369  010652  100001                        BPL     MBDM2B
  (1)  010654                         MBDM2A:
  (2)  010654  104000                        EMT                     ;CC'S INCORRECT
 8370  010656  022700  000001          MBDM2B: CMP     #1,R0
 8371  010662  001401                        BEQ     MBDM2C
  (2)  010664  104000                        EMT                     ;REGISTER NOT INCREMENTED BY ONE
 8372  010666  112720  000252          MBDM2C: MOVB    #252,(R0)+      ;TRY DESTINATION MODE 2 W/ODD BYTE
 8373  010672  102402                        BVS     MBDM2D
 8374  010674  001401                        BEQ     MBDM2D
 8375  010676  100401                        BMI     MBDM2E
  (1)  010700                         MBDM2D:
  (2)  010700  104000                        EMT                     ;CC'S NOT SET CORRECT
 8376  010702  022700  000002          MBDM2E: CMP     #2,R0
 8377  010706  001401                        BEQ     MBDM2F
  (2)  010710  104000                        EMT                     ;REGISTER NOT INCREMENTED BY ONE
 8378  010712  022737  125125  000000  MBDM2F: CMP     #125125,@#0     ;CHECK DATA
 8379  010720  001401                        BEQ     TS152
  (3)  010722  104000                        EMT                     ;DESTINATION DATA INCORRECT
 8380
 8381                                  ;*****************************************************************************
 8382                                  ;
 8383                                  ;       THIS TEST VERIFIES MOV DESTINATION MODE 3. R0 IS USED TO PICK UP
 8384                                  ;AN ADDRESS AT LOC. 400. LOC 400 POINTS TO LOC. 0 THE EFFECTIVE DEST. ADDR.. ALSO, MOVB
 8385                                  ;INST. ARE USED W/ EVEN AND ODD BYTES TO CHECK MOV BYTES INST AND MODE 37 DESTINATIONS.
 8386                                  ;*****************************************************************************
  (2)                                 ;TEST 152       TEST MOV(B) DESTINATION MODE 3
```

```
 (3)                                  ;*******************************************************************************
 (2)  010724                          TS152:
8387  010724  012700  000400                  MOV     #400,R0          ;R0=400
8388  010730  005010                          CLR     (R0)             ;LOC. 400 POINTS TO LOC. 0
8389  010732  005037  000000                  CLR     @#0              ;LOC. 0=0
8390  010736  012730  125252                  MOV     #125252,@(R0)+   ;TRY MOV DESTINATION MODE 2
8391  010742  102402                          BVS     MDM3A            ;BR TO ERROR IF V SET
8392  010744  001401                          BEQ     MDM3A            ;BR TO ERROR IF Z SET
8393  010746  100401                          BMI     MDM3B
 (1)  010750                          MDM3A:
 (2)  010750  104000                          EMT                      ;CC'S INCORRECT
8394  010752  022700  000402          MDM3B:  CMP     #402,R0          ;CHECK DEST. MODE REGISTER
8395  010756  001401                          BEQ     MDM3C
 (2)  010760  104000                          EMT                      ;REGISTER NOT INCREMENTED BY 2
8396  010762  022737  125252  000000  MDM3C:  CMP     #125252,@#0      ;CHECK DESTINATION DATA
8397  010770  001401                          BEQ     MDM3D
 (2)  010772  104000                          EMT                      ;DESTINATION DATA INCORRECT
8398  010774  112737  000125  000000  MDM3D:  MOVB    #125,@#0         ;TRY MOVB DESTINATION MODE Z EVEN BYTE
8399  011002  022737  125125  000000          CMP     #125125,@#0      ;CHECK DATA
8400  011010  001401                          BEQ     MDM3E
 (2)  011012  104000                          EMT                      ;DESTINATION DATA INCORRECT
8401  011014  112737  000525  000001  MDM3E:  MOVB    #525,@#1         ;TRY MOVB DESTINATION MODE 2 ODD BYTE
8402  011022  022737  052525  000000          CMP     #52525,@#0       ;CHECK DATA
8403  011030  001401                          BEQ     TS153
 (3)  011032  104000                          EMT                      ;
8404
8405                                  ;*******************************************************************************
8406                                  ;
8407                                  ;       THIS TEST VERIFIES THE MOV DESTINATION MODE 4 INSTRUCTION.
8408                                  ;SOP INSTRUCTIONS ON R0 ARE USED TO CLEAR TARGET LOCATION 0.
8409                                  ;R4 IS USED AS THE MODE 4 ADDRESSING REGISTER, AND
8410                                  ;CONDITIONAL BRANCHES ARE USED TO VERIFY THE DATA.
8411                                  ;
8412                                  ;*******************************************************************************
 (2)                                  ;TEST 153       TEST MOV DESTINATION MODE 4
 (3)                                  ;*******************************************************************************
 (2)  011034                          TS153:
8413  011034  005000                          CLR     R0               ;R0=0
8414  011036  005010                          CLR     (R0)             ;LOC 0=0
8415  011040  012704  000002                  MOV     #2,R4            ;R4=2
8416  011044  012744  012345                  MOV     #12345,-(R4)     ;TRY MOV DEST. MODE 4
8417  011050  102402                          BVS     MDM4A            ;BR TO ERROR IF V-BIT SET
8418  011052  001401                          BEQ     MDM4A            ;BR TO ERROR IF Z-BIT SET
8419  011054  100001                          BPL     MDM4B
 (1)  011056                          MDM4A:
 (2)  011056  104000                          EMT                      ;CC'S NOT CORRECT
8420  011060  005704                  MDM4B:  TST     R4               ;CHECK DECREMENTING OF MODE 4 REG.
8421  011062  001401                          BEQ     MDM4C
 (2)  011064  104000                          EMT                      ;DESTINATION MODE REGISTER NOT DECREMENTED BY 2
8422  011066  022710  012345          MDM4C:  CMP     #12345,(R0)      ;CHECK DESTINATION DATA
8423  011072  001401                          BEQ     TS154
 (3)  011074  104000                          EMT                      ;DESTINATION DATA INCORRECT
8424
8425                                  ;*******************************************************************************
8426                                  ;
8427                                  ;       THIS TEST VERIFIES THE MOVB DESTINATION MODE 4 INSTRUCTION
```

```
                                ;       ON BOTH ODD AND EVEN BYTES. SOP INSTRUCTIONS ON R4 ARE
                                ;       USED TO CLEAR TARGET LOCATION 0. R0 IS USED AS THE MODE 4
                                ;       ADDRESSING REGISTER, AND CMP AND CONDITIONAL BRANCH
                                ;       INSTRUCTIONS ARE USED TO VERIFY THE DATA.
                                ;
                                ;*******************************************************************************
                                ;TEST 154       TEST MOVB DESTINATION MODE 4
                                ;*******************************************************************************
                                TS154:
                                        CLR     R4              ;R4=0
                                        CLR     (R4)            ;LOC. 0=0
                                        MOV     #2,R0           ;R0 = 2
                                        MOVB    #125125,-(R0)   ;TRY MOVB DEST. MODE 4-ODD BYTE
                                        CMP     R0,#1           ;CHECK THAT DEST. REG. WAS DECREMENTED
                                        BEQ     MBDM4A
                                        EMT                     ;DESTINATION REG. NOT DECREMENTED BY 1
                                MBDM4A: CMP     (R4),#52400     ;CHECK DEST. DATA
                                        BEQ     MBDM4B
                                        EMT                     ;DEST. DATA NOT CORRECT
                                MBDM4B: MOVB    #125125,-(R0)   ;TRY MOVB DEST. MODE 4--EVEN BYTE
                                        BVS     MBDM4C          ;BR. TO ERROR IF V-BIT SET
                                        BEQ     MBDM4C          ;BR TO ERROR IF Z-BIT SET
                                        BPL     MBDM4D
                                MBDM4C:
                                        EMT                     ;COND. CODES INCORRECT
                                MBDM4D: TST     R0              ;CHECK MODE 4 DEST. REGISTER
                                        BEQ     MBDM4E
                                        EMT                     ;DESTINATION REG NOT DECREMENTED BY 1
                                MBDM4E: CMP     (R4),#52525     ;CHECK DEST. DATA
                                        BEQ     TS155
                                        EMT                     ;DESTINATION DATA INCORRECT
                                ;*******************************************************************************
                                ;
                                ;       THIS TEST VERIFIES THE MOV DESTINATION MODE 5 AND THE MOVB
                                ;DESTINATION MODE 5 - EVEN BYTE INSTRUCTIONS. R4 IS A
                                ;POINTER TO TARGET LOCATION 0 AND R0 IS SETUP TO
                                ;POINT TO LOCATION 376 FOR THE MOV, AND LOCATION 404 FOR
                                ;THE MOVB INSTRUCTIONS. CMP INSTRUCTIONS ARE USED TO VERIFY
                                ;PROPER ADDRESSING AND DATA.
                                ;
                                ;*******************************************************************************
                                ;TEST 155       TEST MOV DESTINATION MODE 5
                                ;*******************************************************************************
                                TS155:
                                        CLR     R4              ;R4=0
                                        CLR     (R4)            ;LOC. 0 = 0
                                        MOV     #400,R0         ;R0=400
                                        MOV     #4321,@-(R0)    ;TRY MOV DEST. MODE 5
                                        BVS     MDM5A           ;BR TO ERROR IF V-BIT SET
                                        BEQ     MDM5A           ;BR TO ERROR IF Z-BIT SET
                                        BPL     MDM5B
                                MDM5A:
                                        EMT                     ;COND. CODES INCORRECT
                                MDM5B:  CMP     #376,R0         ;CHECK MODE 5 REG. WAS DECREMENTED
                                        BEQ     MDM5C
```

```
  (2)  011216  104000                          EMT                     ;MODE 5 REGISTER NOT DECREMENTED BY 2
8470   011220  022714  004321        MDM5C:   CMP     #4321,(R4)      ;CHECK DEST. DATA
8471   011224  001401                BEQ     MDM5D
  (2)  011226  104000                EMT                             ;DEST. DATA INCORRECT
8472   011230  012700  000406        MDM5D:   MOV     #406,R0         ;R0=406
8473   011234  112750  000377                 MOVB    #377,@-(R0)     ;TRY MOV DEST. MODE 5 --EVEN BYTE
8474   011240  022700  000404                 CMP     #404,R0         ;CHECK MODE 5 REG.
8475   011244  001401                BEQ     MDM5E
  (2)  011246  104000                EMT                             ;MODE 5 REGISTER NOT DECREMENTED BY 2
8476   011250  022714  177721        MDM5E:   CMP     #177721,(R4)    ;CHECK DEST. DATA
8477   011254  001401                BEQ     TS156
  (3)  011256  104000                EMT                             ;DEST. DATA INCORRECT
8478
8479                                ;*****************************************************************************
8480                                ;
8481                                ;       THIS TEST VERIFIES THE MOV DESTINATION MODE 6 AND MOVB - EVEN BYTE
8482                                ;DESTINATION MODE 6 INDTRUCTIONS. R0 IS USED TO SETUP TARGET LOC.0
8483                                ;FOR BOTH TESTS. PATTERNS OF ONES AND ZEROES ARE MOVED INTO LOC.0
8484                                ;BY MODE 6 INSTRUCTIONS, AND CMP INSTRUCTIONS ARE USED TO VERIFY
8485                                ;PROPER ADDRESSING AND DATA.
8486                                ;
8487                                ;*****************************************************************************
  (2)                               ;TEST 156       TEST MOV DESTINATION MODE 6
  (3)                               ;*****************************************************************************
  (2)  011260                       TS156:
8488   011260  005000                          CLR     R0              ;R0=0
8489   011262  005010                          CLR     (R0)            ;LOC. 0=0
8490   011264  005200                          INC     R0              ;R0=1
8491   011266  012760  052525  177777         MOV     #052525,-1(R0)  ;TRY MOV DEST. MODE 6
8492   011274  102402                          BVS     MDM6A           ;BR TO ERROR IF V-BIT SET
8493   011276  001401                          BEQ     MDM6A           ;BR TO ERROR IF Z-BIT SET
8494   011300  100001                          BPL     MDM6B
  (1)  011302                       MDM6A:
  (2)  011302  104000                          EMT                     ;COND. CODES INCORRECT
8495   011304  022700  000001        MDM6B:   CMP     #1,R0           ;CHECK DEST. REGISTER UNALTERED
8496   011310  001401                BEQ     MDM6C
  (2)  011312  104000                EMT                             ;DEST. REGISTER INCORRECTLY ALTERED
8497   011314  022737  052525  000000 MDM6C:   CMP     #52525,@#0      ;CHECK DEST. DATA
8498   011322  001401                BEQ     MDM6D
  (2)  011324  104000                EMT                             ;DEST. DATA INCORRECT
8499   011326  012700  000002        MDM6D:   MOV     #2,R0           ;R0=2
8500   011332  112760  000377  177777        MOVB    #377,-1(R0)     ;TRY MOVB DEST. MODE 6
8501   011340  022700  000002                CMP     #2,R0           ;CHECK DEST. REGISTER UNALTERED
8502   011344  001401                BEQ     MDM6E
  (2)  011346  104000                EMT                             ;DEST. REGISTER INCORRECTLY ALTERED
8503   011350  022737  177525  000000 MDM6E:   CMP     #177525,@#0     ;CHECK DEST. DATA
8504   011356  001401                BEQ     TS157
  (3)  011360  104000                EMT                             ;DEST. DATA INCORRECT
8505
8506                                ;*****************************************************************************
8507                                ;
8508                                ;       THIS TEST VERIFIES THE MOV DESTINATION MODE 7 AND MOVB - ODD BYTE
8509                                ;DESTINATION MODE 7 INSTRUCTIONS. R4 POINTS TO TARGET LOC.0 AND R0
8510                                ;IS USED AS THE MODE 7 ADDRESSING REGISTER. CMP INSTRUCTIONS ARE
8511                                ;USED TO VERIFY PROPER ADDRESSING AND DATA.
8512                                ;
```

```
                                        ;*******************************************************************************
                                        ;TEST 157       TEST MOV DESTINATION MODE 7
                                        ;*******************************************************************************
011362                                  TS157:
011362  005004                                  CLR     R4                      ;R4=0
011364  005014                                  CLR     (R4)                    ;LOC.0=0
011366  012700  000403                          MOV     #403,R0                 ;R0=403
011372  012770  070707  177777                  MOV     #70707,@-1(R0)          ;TRY MOV W/DEST MODE 7
011400  102402                                  BVS     MDM7A                   ;BR. TO ERROR IF V-BIT SET
011402  001401                                  BEQ     MDM7A                   ;BR TO ERROR IF Z-BIT SET
011404  100001                                  BPL     MDM7B
011406                                  MDM7A:
011406  104000                                  EMT                             ;COND. CODES INCORRECT
011410  022700  000403                  MDM7B:  CMP     #403,R0                 ;CHECK DEST. REGISTER
011414  001401                                  BEQ     MDM7C
011416  104000                                  EMT                             ;DEST. REGISTER INCORRECTLY ALTERED
011420  022737  070707  000000          MDM7C:  CMP     #70707,@#0              ;CHECK DEST. DATA
011426  001401                                  BEQ     MDM7D
011430  104000                                  EMT                             ;DEST. DATA INCORRECT
011432  112770  107070  000001          MDM7D:  MOVB    #107070,@1(R0)          ;TRY MOVB W/DEST MODE 7--ODD BYTE
011440  022700  000403                          CMP     #403,R0                 ;CHECK MODE 7 DEST. REG.
011444  001401                                  BEQ     MDM7E
011446  104000                                  EMT                             ;DEST. DATA INCORRECT
011450  022737  034307  000000          MDM7E:  CMP     #34307,@#0              ;CHECK DEST. DATA
011456  001401                                  BEQ     TS160
011460  104000                                  EMT                             ;DESTINATION DATA INCORRECT

                                        ;*******************************************************************************
                                        ;
                                        ;       THIS TEST VERIFIES MODE 4 DOUBLE OPERAND INSTRUCTIONS.
                                        ;THE TEST USES MODE 4 ADDRESSING WITH REGISTER 0 TO MOVE THRU A
                                        ;TABLE OF OPERANDS.  THE TABLE OF OPERANDS AND THE WORK LOCATION IS
                                        ;STORED FOLLOWING THE TEST CODE.  A SERIES OF 5 DOP INSTRUCTIONS UTILIZES
                                        ;THE DATA IN THE TABLE TO CYCLE THE WORK LOCATION THRU A SET OF
                                        ;VALUE.  THE DATA HAS BEEN CHOSEN TO INSURE THAT NO SINGLE ERROR WILL
                                        ;GO UNDETECTED.  WORD AND BYTE INSTRUCTION ACCESSING BOTH EVEN AND
                                        ;ODD ADDRESSES ARE USED IN THE TEST.  THE LISTING SHOWS THE
                                        ;EXPECTED INTERMEDIATE RESULT AS EACH INSTRUCTION IS EXECUTED.
                                        ;
                                        ;*******************************************************************************
                                        ;TEST 160       TEST MODE 4 W/ DOP INSTS.
                                        ;*******************************************************************************
011462                                  TS160:
011462  012700  011526                          MOV     #TBL1,R0                ;INITIALIZE R0
011466  014037  011526                          MOV     -(R0),@#TBL1            ;TBL1=125252
011472  064037  011526                          ADD     -(R0),@#TBL1            ;TBL1=000377
011476  144037  011526                          BICB    -(R0),@#TBL1            ;TBL1=000252
011502  154037  011527                          BISB    -(R0),@#TBL1+1          ;TBL1=125252
011506  024037  011526                          CMP     -(R0),@#TBL1            ;CHECK RESULT
011512  001406                                  BEQ     TS161
011514                                  DOP4:
011514  104000                                  EMT                             ;RESULT OF MODE 4 INSTS. INCORRECT

011516  125252                                  125252
011520  052652                                  52652
011522  053125                                  53125
```

```
011524  125252                          125252
011526  000000                  TBL1:   0

                                ;*******************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 5 DOUBLE OPERAND INSTRUCTIONS.
                                ;THE TEST USES AN ADDRESS TABLE STORED FOLLOWING THE TEST CODE.
                                ;THIS TABLE IS SIMPLY A TABLE OF ADDRESS POINTERS WHICH ADDRESS
                                ;THE DATA TABLE USED IN THE PREVIOUS TEST.  THE TEST IS IDENTICAL TO
                                ;THE PREVIOUS TEST EXCEPT THE DATA IS REFERENCED USING THIS ADDRESS
                                ;TABLE AND MODE 5 ADDRESSING.  (SEE PREVIOUS TEST).
                                ;
                                ;*******************************************************************************
                                ;TEST 161       TEST MODE 5 W/ DOP INSTS.
                                ;*******************************************************************************
011530                          TS161:
011530  012700  011576                  MOV     #TBL2+2,R0      ;INITIALIZE R0
011534  015037  011526                  MOV     @-(R0),@#TBL1   ;TBL1=125252
011540  065037  011526                  ADD     @-(R0),@#TBL1   ;TBL1=000377
011544  145037  011526                  BICB    @-(R0),@#TBL1   ;TBL1=000252
011550  155037  011527                  BISB    @-(R0),@#TBL1+1 ;TBL1=125252
011554  025037  011526                  CMP     @-(R0),@#TBL1   ;CHECK RESULT
011560  001406                          BEQ     TS162
011562                          DOP5:
011562  104000                          EMT                     ;RESULT OF MODE 5 INSTS. INCORRECT
011564  011516                          TBL1-10
011566  011520                          TBL1-6
011570  011521                          TBL1-5
011572  011522                          TBL1-4
011574  011524                  TBL2:   TBL1-2
                                ;*******************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 6 DOUBLE OPERAND INSTRUCTIONS.
                                ;IT USES THE SAME DATA AS THAT USED IN THE MODE 4 TESTS.
                                ;THIS TIME THE DATA IS ACCESSED USING MODE 6.  R0 IS SET
                                ;TO POINT TO THE MIDDLE OF THE TABLE.  THE TABLE IS ACCESSED FROM
                                ;BOTTOM TO TOP BY VARYING THE OFFSET IN THE MODE 6 INSTRUCTIONS.
                                ;THE DATA RESULTS ARE IDENTICAL TO THOSE EXPECTED IN THE MODE 4
                                ;TESTS.
                                ;
                                ;*******************************************************************************
                                ;TEST 162       TEST MODE 6 W/ DOP INSTS.
                                ;*******************************************************************************
011576                          TS162:
011576  012700  011522                  MOV     #TBL1-4,R0      ;INITIALIZE R0
011602  016037  000002  011526          MOV     2(R0),@#TBL1    ;TBL1=125252
011610  066037  000000  011526          ADD     0(R0),@#TBL1    ;TBL1=000377
011616  146037  177777  011526          BICB    -1(R0),@#TBL1   ;TBL1=000252
011624  156037  177776  011527          BISB    -2(R0),@#TBL1+1 ;TBL1=125252
011632  026037  177774  011526          CMP     -4(R0),@#TBL1   ;CHECK RESULT
011640  001401                          BEQ     TS163
011642  104000                          EMT                     ;RESULT OF MODE 6 INSTS. INCORRECT
                                ;*******************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 7 DOUBLE OPERAND INSTRUCTIONS.
                                ;THIS TEST USES THE SAME ADDRESS TABLE AND DATA TABLE USED BY
```

```
                                        ;THE MODE 5 TESTS.  THIS TIME THE DATA IS ACCESSED USING MODE 7.
                                        ;R0 IS SET TO POINT TO THE MIDDLE OF THE ADDRESS TABLE IN THE MODE 5
                                        ;TEST.  THE TABLE IS ACCESSED FROM BOTTOM TO TOP BY VARYING THE OFFSET
                                        ;IN THE MODE 7 INSTRUCTIONS.  THE DATA RESULTS ARE IDENTICAL TO
                                        ;THOSE EXPECTED IN THE MODE 5 TESTS.
                                        ;
                                        ;*****************************************************************************
                                        ;TEST 163       TEST MODE 7 W/ DOP INSTS.
                                        ;*****************************************************************************
(2) 011644                              TS163:
8609 011644  012700  011570                    MOV     #TBL2-4,R0      ;INITIALIZE R0
8610 011650  017037  000004  011526            MOV     @4(R0),@#TBL1   ;TBL1=125252
8611 011656  067037  000002  011526            ADD     @2(R0),@#TBL1   ;TBL1=000377
8612 011664  147037  000000  011526            BICB    @0(R0),@#TBL1   ;TBL1=000252
8613 011672  157037  177776  011527            BISB    @-2(R0),@#TBL1+1                ;TBL1=125252
8614 011700  027037  177774  011526            CMP     @-4(R0),@#TBL1          ;CHECK RESULT
8615 011706  001401                            BEQ     TS164
(3) 011710  104000                             EMT                             ;RESULT OF MODE 7 INSTS INCORRECT
                                        ;*****************************************************************************
                                        ;
                                        ;       THIS TEST VERIFIES THE ROTATE MODE 0 INSTRUCTIONS.
                                        ;R0 IS LOADED WITH A DATA PATTERN, THE C-BIT IS LOADED, AND
                                        ;AN ROL INSTRUCTION IS EXECUTED WITH MODE 0.  THE OPERATION IS CHECKED
                                        ;BY TESTING THE RESULTING DATA AND THE STATE OF THE C AND V BITS.
                                        ;NEXT, THE SAME PROCEDURE IS EXECUTED TO TEST MODE 0 BYTE INSTRUCTIONS.
                                        ;
                                        ;*****************************************************************************
                                        ;TEST 164       TEST ROTATE INSTRUCTIONS OF MODE 0
                                        ;*****************************************************************************
(2) 011712                              TS164:
8625 011712  012700  125252                    MOV     #125252,R0      ;INITIALIZE DATA
8626 011716  000261                            SEC                     ;SET C-BIT
8627 011720  006100                            ROL     R0              ;TRY ROL W/ MODE 0
8628 011722  102004                            BVC     ROT0A           ;CC=0011
8629 011724  103003                            BCC     ROT0A
8630 011726  022700  052525                    CMP     #052525,R0      ;CHECK DATA
8631 011732  001401                            BEQ     ROT0B
(1) 011734                              ROT0A:
(2) 011734  104000                             EMT                     ;ROL MODE 0 FAILED
8632 011736  012700  125252            ROT0B:  MOV     #125252,R0      ;INITIALIZE DATA
8633 011742  000261                            SEC                     ;SET C-BIT
8634 011744  106100                            ROLB    R0              ;TRY ROL W/ MODE 0 EVEN BYTE
8635 011746  102004                            BVC     ROT0C           ;CC=0011
8636 011750  103003                            BCC     ROT0C
8637 011752  022700  125125                    CMP     #125125,R0      ;CHECK DATA
8638 011756  001401                            BEQ     TS165
(2) 011760                              ROT0C:
(3) 011760  104000                             EMT                     ;ROLB MODE 0 FAILED
8639
8640                                    ;*****************************************************************************
8641                                    ;
8642                                    ;       THIS TEST VERIFIES THE ROTATE MODE 1 INSTRUCTIONS.
8643                                    ;THE DATA TO BE ROTATED IS IN LOC 0.  R0 IS USED AS THE
8644                                    ;ADDRESSING REGISTER.  THE C-BIT IS LOADED AND AN ROL IS EXECUTED.
8645                                    ;THE RESULTS ARE CHECKED BY COMPARING THE DATA RESULTS AND TESTING
8646                                    ;THE C AND V BITS.  THIS PROCEDURE IS THEN REPEATED TWICE MORE
```

```
                        ;TO TEST THE BYTE ROTATES.  FIRST ON BYTE 0, THEN ON BYTE 1.
                        ;
                        ;*****************************************************************************
                        ;TEST 165       TEST ROTATE INSTRUCTIONS W/ MODE 1
                        ;*****************************************************************************
                        TS165:
011762  005000                  CLR     R0              ;POINT TO LOC. 0
011764  012710  052525          MOV     #52525,(R0)     ;INITIALIZE DATA
011770  000241                  CLC                     ;CLEAR C-BIT
011772  006110                  ROL     (R0)            ;TRY ROL W/ MODE 1
011774  102005                  BVC     ROT1A           ;CC=1010
011776  103404                  BCS     ROT1A
012000  023727  000000  125252  CMP     @#0,#125252     ;CHECK RESULT
012006  001401                  BEQ     ROT1B
                        ROT1A:
012010                  EMT                             ;ROL MODE 1 FAILED
012010  104000
012012  000261          ROT1B:  SEC
012014  012710  125252          MOV     #125252,(R0)    ;INITIALIZE DATA
012020  106110                  ROLB    (R0)            ;TRY ROLB W/ MODE 1 EVEN BYTE
012022  102005                  BVC     ROT1C           ;CC=1011
012024  103004                  BCC     ROT1C
012026  022737  125125  000000  CMP     #125125,@#0     ;TEST RESULT
012034  001401                  BEQ     ROT1D
                        ROT1C:
012036                  EMT                             ;ROLB W/ MODE 1 EVEN BYTE FAILED
012036  104000
012040  012710  125252  ROT1D:  MOV     #125252,(R0)
012044  005000                  CLR     R0              ;POINT TO ODD BYTE
012046  005200                  INC     R0
012050  000261                  SEC                     ;SET C-BIT
012052  106110                  ROLB    (R0)            ;TRY ROLB W/ MODE 1 ODD BYTE
012054  102005                  BVC     ROT1E           ;CC=0011
012056  103004                  BCC     ROT1E
012060  022737  052652  000000  CMP     #052652,@#0     ;CHECK DATA
012066  001401                  BEQ     TS166
                        ROT1E:
012070                  EMT                             ;ROLB W/ MODE 1 ODD BYTE FAILED
012070  104000
                        ;*****************************************************************************
                        ;
                        ;       THIS TEST VERIFIES MODE 2 ROTATE INSTRUCTIONS.
                        ;THE SAME PROCEDURE AS IN THE OTHER ROTATE TESTS ARE USED.  R0
                        ;IS USED AS THE ADDRESSING REGISTER AND IS CHECKED FOR PROPER
                        ;INCREMENTING.  BYTE INSTRUCTIONS ARE ALSO CHECKED.
                        ;
                        ;*****************************************************************************
                        ;TEST 166       TEST ROTATE INSTRUCTIONS W/ MODE 2
                        ;*****************************************************************************
                        TS166:
012072  005000                  CLR     R0              ;POINT TO LOC 0
012074  012710  173737          MOV     #173737,(R0)    ;INITIALIZE DATA
012100  000241                  CLC                     ;CLEAR C-BIT
012102  006120                  ROL     (R0)+           ;TRY ROL W/ MODE 2
012104  103007                  BCC     ROT2A           ;CHECK C-BIT
012106  022737  167676  000000  CMP     #167676,@#0     ;CHECK DATA
012114  001003                  BNE     ROT2A           ;BRANCH IF RESULT INCORRECT
012116  005300                  DEC     R0              ;TEST R0
```

```
                                        DEC     R0
                                        BEQ     ROT2B
                                ROT2A:
                                        EMT                             ;ROL W/ MODE 2 FAILED
                                ROT2B:  CLR     R0                      ;POINT TO LOC 0
                                        MOV     #4040,(R0)              ;INITIALIZE DATA
                                        CLC                             ;CLEAR C-BIT
                                        ROLB    (R0)+                   ;TRY ROLB W/ MODE 2 EVEN BYTE
                                        BCS     ROT2C                   ;CHECK C-BIT
                                        CMP     #4100,@#0               ;CHECK DATA
                                        BNE     ROT2C                   ;BRANCH IF DATA INCORRECT
                                        DEC     R0                      ;CHECK R0
                                        BEQ     ROT2D
                                ROT2C:
                                        EMT                             ;ROLB W/ MODE 2 EVEN BYTE FAILED
                                ROT2D:  CLR     R0                      ;POINT TO LOC 0
                                        MOV     #4040,(R0)              ;INITIALIZE DATA
                                        INC     R0                      ;POINT TO ODD BYTE OF DATA
                                        SEC                             ;SET C-BIT
                                        ROLB    (R0)+                   ;TRY ROL W/ MODE 2 ODD BYTE
                                        BCS     ROT2E                   ;CHECK C-BIT
                                        CMP     #10440,@#0              ;CHECK DATA
                                        BNE     ROT2E                   ;BRANCH IF DATA INCORRECT
                                        DEC     R0                      ;CHECK R0
                                        DEC     R0
                                        BEQ     TS167
                                ROT2E:
                                        EMT                             ;ROLB W/ MODE 2 ODD BYTE FAILED
                                ;*******************************************************************************
                                ;
                                ;       THIS TEST VERIFIES MODE 3 ROTATE INSTRUCTIONS.
                                ;THIS TEST USES THE SAME PROCEDURES AS IN THE OTHER ROTATE
                                ;TESTS.  THE DATA IS STORED IN LOC. 0 AND IS ADDRESSED USING
                                ;MODE 37.  BYTE ADDRESSING IS ALSO CHECKED FOR EVEN AND ODD BYTES.
                                ;
                                ;*******************************************************************************
                                ;TEST 167       TEST ROTATE INSTRUCTIONS /W MODE 3
                                ;*******************************************************************************
                                TS167:
                                        MOV     #52525,@#0              ;INITIALIZE DATA IN LOC 0
                                        SEC                             ;SET C-BIT
                                        ROL     @#0                     ;TRO ROL W/ MODE 3
                                        BCS     ROT3A                   ;CHECK C-BIT
                                        CMP     #125253,@#0             ;CHECK DATA
                                        BEQ     ROT3B
                                ROT3A:
                                        EMT                             ;ROL W/ MODE 3 FAILED
                                ROT3B:  MOV     #125252,@#0             ;INITIALIZE DATA
                                        CLC                             ;CLEAR C-BIT
                                        ROLB    @#0                     ;TRY ROL W/ MODE 3 EVEN BYTE
                                        BCC     ROT3C                   ;CHECK C-BIT
                                4$:     CMP     @#0,#125124             ;CHECK DATA
                                        BEQ     ROT3D
                                ROT3C:
                                        EMT                             ;ROL W/ MODE 3 EVEN BYTE FAILED
```

```
 8734  012276  012737  125252  000000  ROT3D:  MOV     #125252,@#0     ;INITIALIZE DATA IN LOC. 0
 8735  012304  000261                          SEC                     ;SET C-BIT
 8736  012306  106137  000001                  ROLB    @#1             ;TRY ROL W/ MODE 3 ODD BYTE
 8737  012312  103004                          BCC     ROT3E           ;CHECK C-BIT
 8738  012314  022737  052652  000000          CMP     #052652,@#0     ;CHECK DATA
 8739  012322  001401                          BEQ     TS170
  (2)  012324                          ROT3E:
  (3)  012324  104000                          EMT                     ;ROL W/ MODE 3 ODD BYTE FAILED
 8740
 8741                                ;*******************************************************************************
 8742                                ;
 8743                                ;       THIS TEST VERIFIES MODE 4 ROTATE INSTRUCTIONS.  THE DATA IS
 8744                                ;STORED IN LOC. 0.  R0 IS SET TO 2 AND THE CARRY IS SET.  AN ROL MODE 4
 8745                                ;IS USED TO ROTATE LOCATION 0 USING R0.  THE DATA IS CHECKED
 8746                                ;AND THE C AND V BITS ARE TESTED.  THE PROPER DECREMENTING OF
 8747                                ;R0 IS VERIFIED.
 8748                                ;
 8749                                ;*******************************************************************************
  (2)                               ;TEST 170       TEST MODE 4 W/ ROTATE INSTRUCTIONS
  (3)                               ;*******************************************************************************
  (2)  012326                       TS170:
 8750  012326  012737  070707  000000          MOV     #070707,@#0     ;INITIALIZE DATA IN LOC. 0
 8751  012334  012700  000002                  MOV     #2,R0           ;INITIALIZE R0 AS POINTER
 8752  012340  000261                          SEC                     ;SET C-BIT
 8753  012342  006140                          ROL     -(R0)           ;TRY ROL W/ MODE 4
 8754  012344  103406                          BCS     ROT4            ;CHECK C-BIT
 8755  012346  022737  161617  000000          CMP     #161617,@#0     ;CHECK DATA
 8756  012354  001002                          BNE     ROT4            ;BRANCH IF DATA INCORRECT
 8757  012356  005700                          TST     R0              ;CHECK MODE 4 REGISTER
 8758  012360  001401                          BEQ     TS171
  (2)  012362                          ROT4:
  (3)  012362  104000                          EMT                     ;ROL MODE 4 FAILED
 8759
 8760                                ;*******************************************************************************
 8761                                ;
 8762                                ;       THIS TEST VERIFIES MODE 5 ROTATE INSTRUCTIONS.
 8763                                ;THE DATA IS STORED IN A WORK LOCATION (ROTX) AT THE END OF THE
 8764                                ;TEST CODE.  LOC. 0 IS LOADED WITH THE ADDRESS OF THE DATA (ROTX).
 8765                                ;R0 IS SET TO 2.  THE CARRY IS CLEARED AND A MODE 5 ROL
 8766                                ;IS EXECUTED USING R0 AS AN ADDRESSING REGISTER.  THE DATA IS
 8767                                ;CHECKED, THE C AND V BITS TESTED, AND R0 CHECKED FOR PROPER
 8768                                ;DECREMENTING.
 8769                                ;
 8770                                ;*******************************************************************************
  (2)                               ;TEST 171       TEST MODE 5 W/ ROTATE INSTRUCTIONS
  (3)                               ;*******************************************************************************
  (2)  012364                       TS171:
 8771  012364  012737  012430  000000          MOV     @#ROTX,@#0      ;MOVE POINTER TO LOC. 0
 8772  012372  012700  000002                  MOV     #2,R0           ;SET MODE 5 REG. TO LOC. 0
 8773  012376  012767  107070  000024          MOV     #107070,ROTX    ;INITIALIZE DATA
 8774  012404  000241                          CLC                     ;CLEAR C-BIT
 8775  012406  006150                          ROL     @-(R0)          ;TRY ROL W/ MODE 5
 8776  012410  103006                          BCC     ROT5            ;CHECK C-BIT
 8777  012412  022737  016160  012430          CMP     #016160,@#ROTX  ;CHECK DATA
 8778  012420  001002                          BNE     ROT5            ;BRANCH IF DATA INCORRECT
 8779  012422  005700                          TST     R0              ;CHECK MODE 5 REGISTER
```

```
 8780  012424  001402                        BEQ     TS172
  (2)  012426                          ROT5:
  (3)  012426  104000                        EMT                     ;ROL MODE 5 FAILED
 8781  012430  000000                  ROTX:  0
 8782
 8783                                 ;*****************************************************************************
 8784                                 ;
 8785                                 ;       THIS TEST VERIFIES MODE 6 ROTATE INSTRUCTIONS.
 8786                                 ;IT USES THE SAME PROCEDURE AS THE ABOVE TEST EXCEPT THE
 8787                                 ;ROTATE INSTRUCTION USES MODE 6 ADDRESSING WITH REGISTER 7.
 8788                                 ;THE DATA IS STILL OPERATED ON IN LOC. ROTX (SEE PREVIOUS TEST).
 8789                                 ;
 8790                                 ;*****************************************************************************
  (2)                                 ;TEST 172       TEST MODE 6 W/ ROTATE INSTRUCTIONS
  (3)                                 ;*****************************************************************************
  (2)  012432                         TS172:
 8791  012432  012737  125252  012430        MOV     #125252,@#ROTX  ;INITIALIZE DATA
 8792  012440  000261                        SEC                     ;SET C-BIT
 8793  012442  006167  177762                ROL     ROTX            ;TRY ROL W/ MODE 6
 8794  012446  103004                        BCC     ROT6            ;CHECK C-BIT
 8795  012450  022737  052525  012430        CMP     #52525,@#ROTX   ;CHECK DATA
 8796  012456  001401                        BEQ     TS173
  (2)  012460                          ROT6:
  (3)  012460  104000                        EMT                     ;ROL W/ MODE 6 FAILED
 8797
 8798                                 ;*****************************************************************************
 8799                                 ;
 8800                                 ;THIS TEST VERIFIES MODE 7 ROTATE INSTRUCTIONS.
```

K6



```
                                        ;THE DATA IS SET IN LOC. ROTX, (SEE PREVIOUS TEST).  THE ROL INSTRUCTION
                                        ;ADDRESSES IT INDIRECTLY USING MODE 7 AND INDIRECT ADDRESS LOCATION
                                        ;(ROTXAD) FOLLOWING THE TEST CODE.
                                        ;
                                        ;*******************************************************************************
                                        ;TEST 173        TEST MODE 7 W/ ROTATE INSTRUCTIONS
                                        ;*******************************************************************************
012462                                  TS173:
012462  012737  052525  012430                  MOV     #52525,@#ROTX   ;INITIALIZE DATA
012470  012737  012430  012520                  MOV     #ROTX,@#ROTXAD  ;INITIALIZE ADDRESS POINTER
012476  000241                                  CLC                     ;CLEAR C-BIT
012500  006177  000014                          ROL     @ROTXAD         ;TRY ROL W/ MODE 7
012504  103404                                  BCS     ROT7            ;CHECK C-BIT
012506  023727  012430  125252                  CMP     @#ROTX,#125252  ;CHECK DATA
012514  001402                                  BEQ     TS174
012516                                  ROT7:
012516  104000                                  EMT                     ;ROL W/ MODE 7 FAILED
012520  000000                          ROTXAD: 0


                                        ;*******************************************************************************
                                        ;
                                        ;       THIS TEST VERIFIES MODE 0 SWAB INSTRUCTION.  R0 IS SET TO
                                        ;177400.  A SWAB MODE 0 IS EXECUTED AND THE CONDITIONAL BRANCH
                                        ;IS USED TO CHECK THE SIGN OF THE RESULT.  ALSO, A COMPARISON
                                        ;IS MADE TO CHECK THE DATA RESULTS.
                                        ;
                                        ;*******************************************************************************
                                        ;TEST 174        TEST MODE 0 W/ SWAB INST.
                                        ;*******************************************************************************
012522                                  TS174:
012522  012700  177400                          MOV     #177400,R0      ;MOVE TEST PATTERN TO R0
012526  000300                                  SWAB    R0              ;TRY SWAB MODE 0
012530  100401                                  BMI     SB0
012532  104000                                  EMT                     ;SWAB DID NOT SET CC'S CORRECT
012534  022700  000377                  SB0:    CMP     #377,R0         ;CHECK RESULT
012540  001401                                  BEQ     TS175
012542  104000                                  EMT                     ;RESULT OF SWAB MODE 0 FAILED


                                        ;*******************************************************************************
                                        ;
                                        ;       THIS TEST VERIFIES MODE 1 SWAB INSTRUCTION.  THE TEST
                                        ;PATTERN IS MOVED TO LOC 0.  R0 IS CLEARED AND USED AS THE ADDRESSING
                                        ;REGISTER IN THE MODE 1 SWAB.  THE DATA RESULTS ARE CHECKED WITH
                                        ;A COMPARE.
                                        ;
                                        ;*******************************************************************************
                                        ;TEST 175        TEST MODE 1 W/ SWAB INST
                                        ;*******************************************************************************
012544                                  TS175:
012544  012737  125652  000000                  MOV     #125652,@#0     ;MOVE TEST PATTERN TO LOC. 0
012552  005000                                  CLR     R0              ;R0=0
012554  000310                                  SWAB    (R0)            ;TRY SWAB MODE 1
012556  022737  125253  000000                  CMP     #125253,@#0     ;CHECK RESULT
012564  001401                                  BEQ     TS176
012566  104000                                  EMT                     ;RESULT OF SWAB MODE 1 FAILED
```

```
                                        ;*******************************************************************************
                                        ;
                                        ;       THIS TEST VERIFIES MODE 2 SWAB INSTRUCTION.  THE TEST
                                        ;PATTERN IS MOVED TO LOC 0.  R0 IS CLEARED AND USED AS THE MODE
                                        ;2 ADDRESSING REGISTER.  THE RESULTS ARE CHECKED WITH A COMPARE.
                                        ;R0 IS CHECKED FOR PROPER DECREMENTING.
                                        ;
                                        ;*******************************************************************************
                                        ;TEST 176       TEST MODE 2 W/ SWAB INST
                                        ;*******************************************************************************
 (2)  012570                            TS176:
 (3)  012570  012737  125152  000000            MOV     #125152,@#0     ;MOVE TEST PATTERN TO LOC. 0
 (2)  012576  005000                            CLR     R0              ;R0=0
 (3)  012600  000320                            SWAB    (R0)+           ;TRY SWAB MODE 2
 (2)  012602  022737  065252  000000            CMP     #65252,@#0      ;CHECK RESULT
 (3)  012610  001401                            BEQ     SB2
 (2)  012612  104000                            EMT                     ;RESULT OF SWAB MODE 0 FAILED
 (3)  012614  162700  000002            SB2:    SUB     #2,R0           ;CHECK EFFECT OF REG.
 (2)  012620  001401                            BEQ     TS177
 (3)  012622  104000                            EMT                     ;REGISTER VALUE INCORRECT


                                        ;*******************************************************************************
                                        ;
                                        ;       THIS TEST VERIFIES MODE 3 SWAB INSTRUCTION.  THE TEST
                                        ;PATTERN IS MOVED TO LOC 0.  A MODE 3 SWAB INSTRUCTION IS EXECUTED
                                        ;USING R7 AS THE ADDRESSING REGISTER.  A COMPARE VERIFIES THE
                                        ;DATA RESULTS.
                                        ;
                                        ;*******************************************************************************
                                        ;TEST 177       TEST MODE 3 W/SWAB INST.
                                        ;*******************************************************************************
 (2)  012624                            TS177:
 (3)  012624  012737  000377  000000            MOV     #377,@#0        ;MOVE TEST PATTERN TO LOC. 0
 (2)  012632  000337  000000                    SWAB    @#0             ;TRY SWAB W/ MODE 3
 (3)  012636  022737  177400  000000            CMP     #177400,@#0     ;CHECK RESULT
 (2)  012644  001401                            BEQ     TS200
 (3)  012646  104000                            EMT                     ;RESULT OF SWAB INCORRECT


                                        ;*******************************************************************************
                                        ;
                                        ;       THIS TEST VERIFIES MODE 4 SWAB INSTRUCTIONS.  THE DATA
                                        ;IS MOVED TO LOC 0.  R0 IS SET TO 2 AND USED AS THE MODE 4 ADDRESSING
                                        ;REGISTER.  THE DATA IS CHECKED WITH A COMPARE AND R0 IS CHECKED
                                        ;FOR PROPER DECREMENTING.
                                        ;
                                        ;*******************************************************************************
                                        ;TEST 200       TEST MODE 4 W/ SWAB INST
                                        ;*******************************************************************************
 (2)  012650                            TS200:
 (3)  012650  012737  125652  000000            MOV     #125652,@#0     ;MOVE TEST PATTERN TO LOC. 0
 (2)  012656  012700  000002                    MOV     #2,R0           ;SET UP REGISTER POINTER
 (3)  012662  000340                            SWAB    -(R0)           ;TRY SWAB MODE 4
```

```
                                CMP     #125253,#00     ;CHECK RESULT
                                BEQ     SB4
                                EMT                     ;RESULT OF SWAB INCORRECT
                        SB4:    TST     R0              ;CHECK EFFECT ON REG.
                                BEQ     TS201
                                EMT                     ;REGISTER VALUE INCORRECT


                        ;*****************************************************************************
                        ;
                        ;       THIS TEST VERIFIES MODE 5 SWAB INSTRUCTION.  THE TEST USES
                        ;TWO LOCATIONS FOLLOWING THE TEST CODE.  SB5X HOLDS THE DATA;
                        ;SB5XAD IS A POINTER TO THE DATA LOCATION.  THE DATA IS MOVED TO
                        ;SB5X AND R0 IS SET TO TWO PLUS THE ADDRESS OF SB5XAD.  FOLLOWING
                        ;THE MODE 5 SWAB SB5X IS CHECKED FOR THE PROPER DATA.  R0 IS
                        ;CHECKED TO SEE THAT IT WAS DECREMENTED PROPERLY.
                        ;
                        ;*****************************************************************************
                        ;TEST 201       TEST MODE 5 W/ SWAB INST.
                        ;*****************************************************************************
                        TS201:
                                MOV     #SB5XAD+2,R0    ;SET UP POINTER TO WORK LOCATION
                                MOV     #125125,SB5X    ;MOVE PATTERN TO WORK LOCATION
                                SWAB    @-(R0)          ;TRY SWAB MODE 5
                                CMP     #52652,SB5X     ;CHECK RESULT
                                BEQ     SB5A
                                EMT                     ;RESULT OF SWAB INCORRECT
                        SB5A:   CMP     R0,#SB5XAD      ;CHECK RESULT OF REG.
                                BEQ     TS202
                        SB5:
                                EMT                     ;REGISTER VALUE INCORRECT
                        SB5X:   0                       ;WORK LOCATION
                        SB5XAD: SB5X


                        ;*****************************************************************************
                        ;
                        ;       THIS TEST VERIFIES MODE 6 SWAB INSTRUCTION.  THIS TEST
                        ;USES A WORK LOCATION (SB6X) FOLLOWING THE TEST CODE.  TEST DATA
                        ;IS LOADED INTO THE WORK LOCATION.  R0, THE ADDRESSING REGISTER
                        ;IS LOADED WITH 6 LESS THEN THE ADDRESS OF THE WORK LOCATION.
                        ;THE MODE 6 SWAB IS EXECUTED WITH A +6 OFFSET.  THE DATA IS
                        ;VERIFIED WITH A COMPARE.
                        ;
                        ;*****************************************************************************
                        ;TEST 202       TEST MODE 6 W/ SWAB INST.
                        ;*****************************************************************************
                        TS202:
                                MOV     #125125,SB6X    ;MOVE PATTERN TO WORK LOCATION
                                MOV     #SB6X-6,R0      ;MOVE OFFSET POINTER TO R0
                                SWAB    6(R0)           ;TRY SWAB W/ MODE 6
                                CMP     #52652,6(R0)    ;CHECK RESULT
                                BEQ     TS203
                        SB6:
                                EMT                     ;RESULT OF SWAB INCORRECT
                        SB6X:   0                       ;WORK LOCATION
```

N6



```
                                        ;*******************************************************************************
                                        ;
                                        ;       THIS TEST VERIFIES MODE 7 SWAB INSTRUCTION.  THIS TEST
                                        ;USES TWO LOCATIONS FOLLOWING THE TEST CODE:  A WORK LOCATION
                                        ;(SB7X) AND A POINTER TO THE WORK LOCATION (SB7XAD).  DATA IS MOVED
                                        ;TO THE WORK LOCATION.  R0 IS LOADED WITH 72 LESS THAN THE ADDRESS
                                        ;OF THE ADDRESS POINTER.  THE DATA IS SWAB'ED USING A MODE 7
                                        ;INSTRUCTION WITH AN OFFSET OF +72.  THE DATA IS VERIFIED WITH A
                                        ;COMPARE.
                                        ;
                                        ;*******************************************************************************
                                        ;TEST 203       TEST MODE 7 W/ SWAB INST.
                                        ;*******************************************************************************
 (2)  013000                            TS203:
 (2)  013000  012767  177400  000022            MOV     #177400,SB7X    ;MOVE PATTERN TO WORK LOCATION
 (2)  013006  012700  012740                    MOV     #SB7XAD-72,R0   ;MOVE OFFSET POINTER TO R0
 (2)  013012  000370  000072                    SWAB    @72(R0)         ;TRY SWAB MODE 7
 (2)  013016  027027  000072  000377            CMP     @72(R0),#377    ;CHECK RESULTS
 (2)  013024  001403                            BEQ     TS204
 (2)  013026                            SB7:
 (3)  013026  104000                            EMT                     ;RESULT OF SWAB INCORRECT
 (2)  013030  000000                    SB7X:   0                       ;WORK LOCATION
 (2)  013032  013030                    SB7XAD: SB7X                    ;POINTER TO WORK LOCATION


                                        ;*******************************************************************************
                                        ;
                                        ;       THIS TEST VERIFIES ALL LEGAL MODES OF THE JMP INSTRUCTION.
                                        ;BECAUSE OF THE NATURE OF THE INSTRUCTION UNDER TEST, THIS TEST
                                        ;UTILIZES SEVERAL DIFFERENT TECHNIQUES.  THE CODE IS NOT EXECUTED
                                        ;IN A LINEAR FASHION.  THE DIFFERENT MODES ARE EXECUTED IN ORDER
                                        ;FROM 1-7; HOWEVER, THE CODE IS ARRANGED SO THAT CONTROL LEAP
                                        ;FROGS THRU THE TEST CODE.  THE ORDER OF APPEARANCE OF THE CODE
                                        ;IS:
                                        ;       JMP MODE 1
                                        ;       JMP MODE 3
                                        ;       JMP MODE 2
                                        ;       JMP MODE 4
                                        ;       JMP MODE 6
                                        ;       JMP MODE 5
                                        ;       JMP MODE 7
                                        ;AN INTERNAL SEQUENCE TEST (JMPSEQ) IS USED TO INSURE THAT THE
                                        ;JUMPS ARE OCCURRING IN THE PROGRAMMED SEQUENCE.
                                        ;       THE TEST IS MADE UP OF SEVERAL BLOCKS OF CODE.  EACH CODE
                                        ;BEGINS WITH A LABEL WHICH INDICATES THE MODE BEING EXECUTED IN
                                        ;THAT BLOCK.  A SIMPLE PROCEDURE IS FOLLOWED IN EACH BLOCK.  FOR
                                        ;EXAMPLE THE CODE BEGINNING AT JMP3 WILL FIRST COMPARE THE RESULTS
                                        ;OF THE PREVIOUS MODE 2 JUMP.  (ANY REGISTER CHANGES ARE VERIFIED
                                        ;AND THE SEQUENCE CHECK IS MADE).  THEN THE REGISTERS ARE SETUP
                                        ;FOR A MODE 3 JUMP TO THE NEXT TEST BLOCK (HERE, JMP4), THE SEQUENCE
                                        ;CHECKER IS UPDATED AND THE JUMP IS EXECUTED.
                                        ;       IF A FAILURE OCCURS, THE SEQUENCE CHECKER WILL ASSIST IN
                                        ;DETERMINING JUST WHICH MODE FAILED.  IF THE SEQUENCE IS CORRECT
                                        ;THEN THE ERROR DETECTED WAS A MODE FAILURE (E.G. FAILURE OF THE
```

```
                                        ;REGISTER TO BE INCREMENTED IN MODE 2 JUMP.)
                                        ;
                                        ;*****************************************************************************
                                        ;TEST 204       TEST THE JMP INSTRUCTION IN ALL MODES
                                        ;*****************************************************************************
 (2) 013034                             TS204:
8985 013034  005067  000240                    CLR     JMPSEQ          ;ESTABLISH A SEQUENCE CHECKER
8986 013040  012700  013104                    MOV     #JMP2,R0        ;SET R0=JUMP TARGET
8987 013044  000110                            JMP     (R0)            ;TRY JMP MODE 1
8988 013046  022700  013050            JMP3:   CMP     #.+2,R0         ;CHECK RESULT OF MODE 2 JUMP
8989 013052  001401                            BEQ     JMP3A
 (2) 013054  104000                            EMT                     ;REGISTER VALUE AFTER JMP MODE 2 INCORRECT
8990 013056  026727  000216  000001    JMP3A:  CMP     JMPSEQ,#1       ;MAKE SURE JMPS ARE IN SEQUENCE: JMPSEQ=1?
8991 013064  001401                            BEQ     JMP3B
 (2) 013066  104000                            EMT                     ;SHOULD BE HERE FROM JMP MODE 2 ONLY
8992 013070  012700  013102            JMP3B:  MOV     #IJMP4,R0       ;POINT R0 TO INDIRECT JMP ADDR.
8993 013074  005267  000200                    INC     JMPSEQ          ;UPDATE SEQUENCE CHECKER
8994 013100  000130                            JMP     @(R0)+          ;TRY JMP MODE 3
8995 013102  013126                    IJMP4:  JMP4                    ;ADDRESS INDIRECT JUMP
8996
8997 013104  005767  000170            JMP2:   TST     JMPSEQ          ;CHECK THAT JMPS ARE IN SEQUENCE: JMPSEQ=0?
8998 013110  001401                            BEQ     JMP2A
 (2) 013112  104000                            EMT                     ;SHOULD BE HERE FROM JMP MODE 1 ONLY
8999 013114  005267  000160            JMP2A:  INC     JMPSEQ          ;UPDATE SEQUENCE CHECKER
9000 013120  012700  013046                    MOV     #JMP3,R0        ;SET R0=JUMP TARGET
9001 013124  000120                            JMP     (R0)+           ;TRY A JUMP MODE 2 TO "JMP3"
9002 013126  022700  013104            JMP4:   CMP     #IJMP4+2,R0     ;CHECK RESULT OF REGISTER IN MODE 3 JUMP
9003 013132  001401                            BEQ     JMP4A
 (2) 013134  104000                            EMT                     ;REGISTER VALUE AFTER MODE 3 JUMP INCORRECT
9004 013136  022767  000002  000134    JMP4A:  CMP     #2,JMPSEQ       ;CHECK JUMP SEQUENCE: JMPSEQ=2?
9005 013144  001401                            BEQ     JMP4B
 (2) 013146  104000                            EMT                     ;SHOULD BE ONLY FROM MODE 3 JUMP
9006 013150  012700  013212            JMP4B:  MOV     #JMP5+2,R0      ;SET UP POINTER TO JUMP TARGET
9007 013154  005267  000120                    INC     JMPSEQ          ;UPDATE SEQUENCE CHECKER
9008 013160  000140                            JMP     -(R0)           ;TRY JUMP MODE 4 TO "JMP4"
9009
9010 013162  022767  000004  000110    JMP6:   CMP     #4,JMPSEQ       ;CHECK THAT JUMPS ARE IN SEQUENCE: JMPSEQ=4?
9011 013170  001401                            BEQ     JMP6A
 (2) 013172  104000                            EMT                     ;SHOULD BE HERE ONLY FROM MODE 5 JUMP
9012 013174  012700  013634            JMP6A:  MOV     #JMP7+376,R0    ;SET UP OFFSET POINTER TO JUMP TARGET
9013 013200  005267  000074                    INC     JMPSEQ          ;UPDATE JUMP SEQUENCE
9014 013204  000160  177402                    JMP     -376(R0)        ;TRY MODE 6 JUMP
9015
9016 013210  022767  000003  000062    JMP5:   CMP     #3,JMPSEQ       ;CHECK THAT JUMPS ARE IN SEQUENCE: JMPSEQ=3?
9017 013216  001401                            BEQ     JMP5A
 (2) 013220  104000                            EMT                     ;SHOULD ONLY BE HERE FROM MODE 4 JUMP
9018 013222  012700  013236            JMP5A:  MOV     #IJMP5+2,R0     ;SET UP POINTER TO INDIRECT JUMP ADDR.
9019 013226  005267  000046                    INC     JMPSEQ          ;UPDATE JUMP SEQUENCE
9020 013232  000150                            JMP     @-(R0)          ;TRY JUMP MODE 5 TO "JMP6"
9021 013234  013162                    IJMP5:  JMP6                    ;INDIRECT ADDRESS POINTER
9022
9023 013236  022767  000005  000034    JMP7:   CMP     #5,JMPSEQ       ;CHECK JUMPS IN SEQUENCE: JMPSEQ=5?
9024 013244  001401                            BEQ     JMP7A
 (2) 013246  104000                            EMT                     ;SHOULD ONLY BE HERE FROM MODE 6 JUMP
9025 013250  012700  013274            JMP7A:  MOV     #IJMP+10,R0     ;SET UP OFFSET POINTER TO INDIRECT ADDR.
9026 013254  005267  000020                    INC     JMPSEQ          ;UPDATE JUMP SEQUENCE
```

```
 9027  013260  000170  177770            JMP     @-10(R0)        ;TRY MODE 7 JUMP
 9028  013264  013266                IJMP:  JMPCK                   ;INDIRECT ADDRESS
 9029
 9030  013266  026727  000006  000006  JMPCK: CMP     JMPSEQ,#6       ;CHECK JUMPS IN SEQUENCE: JMPSEQ
 9031  013274  001402                        BEQ     TS205
  (3)  013276  104000                        EMT                     ;SHOULD ONLY BE HERE FROM MODE 6 JUMP
 9032  013300  000000                JMPSEQ: 0
 9033
 9034                                ;*****************************************************************************
 9035                                ;
 9036                                ;       THIS TEST VERIFIES ALL LEGAL MODES OF THE JSR INSTRUCTION.
 9037                                ;THE CONCEPT OF LEAP FROGGING AND SEQUENCE CHECKING (JSRSEQ) IS
 9038                                ;IDENTICAL TO THAT USED IN JMP TEST (SEE PREVIOUS TEST).  EACH
 9039                                ;BLOCK OF CODE VERIFIES THE PREVIOUS JSR BY CHECKING THE SEQUENCE.
 9040                                ;CHECKING THAT THE PC WAS SAVED IN THE SPECIFIED REGISTER, CHECKING
 9041                                ;THAT THE SP WAS DECREMENTED, CHECKING THAT THE REGISTER WAS
 9042                                ;SAVED ON THE STACK, AND FINALLY CHECKING THAT ANY MODE ADDRESS
 9043                                ;REGISTER ALTERATIONS (E.G. INCREMENT REGISTER IN MODE 2) WERE
 9044                                ;SUCCESSFUL.  R1 IS USED AS THE REGISTER IN ALL JSR INSTRUCTIONS.
 9045                                ;       IF A FAILURE OCCURS, THE SEQUENCE CHECKER WILL ASSIST IN
 9046                                ;DETERMINING JUST WHICH MODE FAILED.  IF THE SEQUENCE IS CORRECT
 9047                                ;THEN THE ERROR DETECTED WAS A FUNCTIONAL FAILURE (E.G., INCORRECT
 9048                                ;REGISTER SAVED).
 9049                                ;
 9050                                ;*****************************************************************************
  (2)                               ;TEST 205       TEST JSR INSTRUCTION W/ ALL MODES
  (3)                               ;*****************************************************************************
  (2)  013302                       TS205:
 9051  013302  000402                        BR      JSR1
 9052  013304  000137  013666        JSR0:   JMP     @#JSRCK1
 9053
 9054  013310  012706  001000        JSR1:   MOV     #STBOT,R6       ;SET STACK POINTER
 9055  013314  012700  013406                MOV     #JSR2,R0        ;SET TARGET ADDRESS
 9056  013320  005037  013646                CLR     @#JSRSEQ        ;INITIALIZE SEQUENCE CHECKER
 9057  013324  005001                        CLR     R1              ;INITIALIZE R1
 9058  013326  005101                        COM     R1
 9059  013330  004110                        JSR     R1,(R0)         ;TRY JSR MODE 1
 9060                                                                ; TO SCOPE: REPLACE THE MOVE INSTRUCTION   <****
 9061                                                                ;           FOLLOWING W/ 774               <****
 9062  013332                        JSR1A:
  (2)  013332  104000                        EMT                     ;JSR MODE 1 FAILED
 9063
 9064  013334  022737  000001  013646  JSR3:   CMP     #1,@# JSRSEQ    ;CHECK SEQUENCE: JSRSEQ=1?
 9065  013342  001014                        BNE     JSR3A           ;BRANCH IF OUT OF SEQUENCE
 9066  013344  020127  013462                CMP     R1,#JSR4        ;PROPER PC SAVED?
 9067  013350  001011                        BNE     JSR3A           ;BRANCH IF PC WRONG
 9068  013352  022706  000776                CMP     #STBOT-2,R6     ;STACK POINTER DECREMENTED?
 9069  013356  001006                        BNE     JSR3A           ;BRANCH IF SP WRONG
 9070  013360  022716  125252                CMP     #125252,(R6)    ;REG SAVED ON STACK?
 9071  013364  001003                        BNE     JSR3A           ;BRANCH IF REG. NOT SAVED
 9072  013366  022700  013336                CMP     #JSR3+2,R0      ;MODE 2 INCREMENT CORRECT?
 9073  013372  001401                        BEQ     JSR3B
  (1)  013374                        JSR3A:
  (2)  013374  104000                        EMT                     ;JSR MODE 3 MALFUNCTIONED
 9074  013376  005237  013646        JSR3B:  INC     @#JSRSEQ        ;UPDATE SEQUENCE CHECKER
 9075  013402  004137  013462                JSR     R1,@#JSR4       ;TRY JSR MODE 4
```

```
 9076
 9077  013406  005737  013646          JSR2:   TST     @#JSRSEQ        ;CHECK SEQUENCE: JSRSEQ=0?
 9078  013412  001011                          BNE     JSR2A           ;BRANCH IF OUT OF SEQUENCE
 9079  013414  020127  013332                  CMP     R1,#JSR1A       ;PROPER PC SAVED?
 9080  013420  001006                          BNE     JSR2A           ;BRANCH IF PC WRONG
 9081  013422  022706  000776                  CMP     #STBOT-2,R6     ;R6 DECREMENT?
 9082  013426  001003                          BNE     JSR2A           ;BRANCH IF R6 IS INCORRECT
 9083  013430  021627  177777                  CMP     (R6),#-1        ;REGISTER SAVED?
 9084  013434  001401                          BEQ     JSR2B
   (1) 013436                          JSR2A:
   (2) 013436  104000                          EMT                     ;JSR MODE 1 MALFUNCTIONED
 9085  013440  012706  001000          JSR2B:  MOV     #STBOT,R6       ;INITIALIZE R6
 9086  013444  012701  125252                  MOV     #125252,R1      ;INITIALIZE R1
 9087  013450  005237  013646                  INC     @#JSRSEQ        ;UPDATE SEQUENCE CHECKER
 9088  013454  012700  013334                  MOV     #JSR3,R0        ;SET TARGET ADDRESS
 9089  013460  004120                          JSR     R1,(R0)+        ;TRY JSR MODE 2
 9090
 9091  013462  022737  000002  013646  JSR4:   CMP     #2,@#JSRSEQ     ;CHECK SEQUENCE: JSRSEQ=2?
 9092  013470  001003                          BNE     JSR4A           ;BRANCH IF OUT OF SEQUENCE
 9093  013472  022701  013406                  CMP     #JSR2,R1        ;PROPER PC SAVED?
 9094  013476  001401                          BEQ     JSR4B
   (1) 013500                          JSR4A:
   (2) 013500  104000                          EMT                     ;JSR MODE 3 MALFUNCTIONED
 9095  013502  005237  013646          JSR4B:  INC     @#JSRSEQ        ;UPDATE SEQUENCE CHECKER
 9096  013506  012700  013554                  MOV     #JSR5+2,R0      ;SET TARGET ADDRESS
 9097  013512  004140                          JSR     R1,-(R0)        ;TRY JSR MODE 4
 9098
 9099  013514  022767  000004  000124  JSR6:   CMP     #4,JSRSEQ       ;CHECK SEQUENCE: JSRSEQ=4?
 9100  013522  001006                          BNE     JSR6A           ;BRANCH IF OUT OF SEQUENCE
 9101  013524  022701  013612                  CMP     #JSR7,R1        ;PROPER PC SAVED?
 9102  013530  001003                          BNE     JSR6A           ;BRANCH IF PC WRONG
 9103  013532  022700  013642                  CMP     #JSR6AD,R0      ;MODE 5 REGISTER CORRECT?
 9104  013536  001401                          BEQ     JSR6B
   (1) 013540                          JSR6A:
   (2) 013540  104000                          EMT                     ;JSR MODE 5 FAILED
 9105  013542  005237  013646          JSR6B:  INC     @#JSRSEQ        ;UPDATE SEQUENCE CHECKER
 9106  013546  004167  000040                  JSR     R1,JSR7         ;TRY JSR MODE 6
 9107  013552  022767  000003  000066  JSR5:   CMP     #3,JSRSEQ       ;CHECK SEQUENCE: JSRSEQ=3?
 9108  013560  001006                          BNE     JSR5A           ;BRANCH IF OUT OF SEQUENCE
 9109  013562  022701  013514                  CMP     #JSR6,R1        ;PROPER PC SAVED?
 9110  013566  001003                          BNE     JSR5A           ;BRANCH IF PC WRONG
 9111  013570  022700  013552                  CMP     #JSR5,R0        ;CHECK MODE 4 REGISTER
 9112  013574  001401                          BEQ     JSR5B
   (1) 013576                          JSR5A:
   (2) 013576  104000                          EMT                     ;JSR MODE 4 MALFUNCTIONED
 9113  013600  005237  013646          JSR5B:  INC     @#JSRSEQ        ;UPDATE SEQUENCE CHECKER
 9114  013604  012700  013644                  MOV     #JSR6AD+2,R0    ;POINT R0 TO TARGET ADDRESS
 9115  013610  004150                          JSR     R1,@-(R0)       ;TRY JSR MODE 5
 9116
 9117  013612  022737  000005  013646  JSR7:   CMP     #5,@#JSRSEQ     ;CHECK SEQUENCE: JSRSEQ=5?
 9118  013620  001003                          BNE     JSR7A           ;BRANCH IF OUT OF SEQUENCE
 9119  013622  022701  013552                  CMP     #JSR5,R1        ;PROPER PC SAVED?
 9120  013626  001401                          BEQ     JSR7B
   (1) 013630                          JSR7A:
   (2) 013630  104000                          EMT                     ;JSR MODE 6 FAILED
 9121  013632  005237  013646          JSR7B:  INC     @#JSRSEQ        ;UPDATE SEQUENCE CHECKER
```

```
 9122  013636  004177  000002                    JSR     R1,@JSRCKAD     ;TRY JSR MODE 7
 9123
 9124  013642  013514                    JSR6AD: JSR6                    ;MODE 5 TARGET ADDRESS
 9125  013644  013650                    JSRCKAD:JSRCK                   ;MODE 7 TARGET ADDRESS
 9126  013646  000000                    JSRSEQ: 0                       ;SEQUENCE CHECKER
 9127
 9128  013650  022767  000006  177770    JSRCK:  CMP     #6,JSRSEQ       ;CHECK SEQUENCE: JSRSEQ=6?
 9129  013656  001003                            BNE     JSRCK1          ;BRANCH IF OUT OF SEQUENCE
 9130  013660  022701  013642                    CMP     #JSR6AD,R1      ;PROPER PC SAVED?
 9131  013664  001401                            BEQ     TS206
  (2)  013666                            JSRCK1:
  (3)  013666  104000                            EMT                     ;JSR MODE 7 MALFUNCTIONED
 9132
 9133
 9134                                    ;*****************************************************************************
 9135                                    ;
 9136                                    ;       THIS TEST VERIFIES THE RTS INSTRUCTION.  THE STACK POINTER
 9137                                    ;IS INITIALIZED AND A TEST PATTERN STORED ON STACK.  R0 IS LOADED
 9138                                    ;WITH RETURN ADDRESS.  AN RTS IS EXECUTED, AND, AT THE TARGET
 9139                                    ;ADDRESS, A CHECK IS MADE THAT R0 WAS PROPERLY RESTORED FROM THE
 9140                                    ;STACK.
 9141                                    ;*****************************************************************************
  (2)                                   ;TEST 206       TEST RTS INSTRUCTION
  (3)                                   ;*****************************************************************************
  (2)  013670                           TS206:
 9142  013670  012706  001000                    MOV     #STBOT,R6       ;INITIALIZE STACK POINTER
 9143  013674  012746  052525                    MOV     #52525,-(R6)    ;INITIALIZE TOP OF STACK
 9144  013700  012700  013710                    MOV     #RTS1,R0        ;INITIALIZE RETURN REGISTER
 9145  013704  000200                            RTS     R0              ;TRY RTS THROUGH R0
 9146                                                                    ; TO SCOPE: REPLACE THE MOVE INSTRUCTION   <****
 9147                                                                    ;           FOLLOWING W/ 770               <****
 9148  013706  104000                            EMT                     ;RTS FAILED
 9149  013710  022700  052525           RTS1:   CMP     #52525,R0       ;CHECK THAT R0 RESTORED FROM STACK
 9150  013714  001401                            BEQ     TS207
  (3)  013716  104000                            EMT                     ;RTS MALFUNCTIONED
 9151                                    ;*****************************************************************************
 9152                                    ;
 9153                                    ;       THESE NEXT FOUR TESTS VERIFY THE FUNCTIONING OF A GROUP
 9154                                    ;OF FOUR INSTRUCTIONS.  THE GROUP CONSISTS OF THE INSTRUCTIONS:
 9155                                    ;MOV, BIC, BIT, AND BIS.  THESE INSTRUCTIONS ARE SIMILAR IN THE
 9156                                    ;WAY THEY EFFECT THE C AND V BITS.  THEY ALL LEAVE THE V-BIT
 9157                                    ;CLEAR AND THE C-BIT UNAFFECTED.
 9158                                    ;       THE TEST PROCEDURE IS AS FOLLOWS:  THE N, Z, AND V BITS
 9159                                    ;ARE LOADED WITH THE COMPLEMENT OF THE EXPECTED RESULTS. THE C-BIT
 9160                                    ;IS LOADED WITH THE DESIRED RESULT.  THE INSTRUCTION IS EXECUTED
 9161                                    ;WITH DIFFERENT DATA PATTERNS AND THE RESULTS ARE VERIFIED WITH
 9162                                    ;A SERIES OF CONDITIONAL BRANCH INSTRUCTIONS.  THE DATA IS CHOSEN
 9163                                    ;TO PRODUCT ALL POSSIBLE COMBINATIONS OF THE C AND V BITS.
 9164                                    ;
 9165                                    ;*****************************************************************************
  (2)                                   ;TEST 207       TEST MOV INSTRUCTION
  (3)                                   ;*****************************************************************************
  (2)  013720                           TS207:
 9166  013720  000277                            SCC                     ;CC=0110
 9167  013722  000251                            .CLN!CLC
 9168  013724  012700  100000                    MOV     #100000,R0      ;CC=1000
```

```
 9169  013730  101402                        BLOS    MOV1
 9170  013732  102401                        BVS     MOV1
 9171  013734  100401                        BMI     MOV2
   (1) 013736                          MOV1:
   (2) 013736  104000                        EMT                     ;MOV DID NOT SET CC'S CORRECTLY
 9172
 9173  013740  000277                  MOV2: SCC                     ;CC=1011
 9174  013742  000244                        CLZ
 9175  013744  012700  000000                MOV     #0,R0           ;CC=0101
 9176  013750  101002                        BHI     MOV3            ;C OR Z = 0?
 9177  013752  102401                        BVS     MOV3            ;V=1?
 9178  013754  100001                        BPL     TS210
   (2) 013756                          MOV3:
   (3) 013756  104000                        EMT                     ;MOV DID NOT SET CC'S CORRECTLY
                                      ;*****************************************************************************
 9179                                 ;TEST 210       TEST BIT INSTRUCTION
   (2)                                ;*****************************************************************************
   (3)
   (2) 013760                         TS210:
 9180  013760  012700  100001                MOV     #100001,R0
 9181  013764  000277                        SCC                     ;CC=0110
 9182  013766  000251                        .CLN!CLC
 9183  013770  032700  100000                BIT     #100000,R0      ;CC=1000
 9184  013774  101402                        BLOS    BITST1
 9185  013776  102401                        BVS     BITST1
 9186  014000  100401                        BMI     BITST2
   (1) 014002                          BITST1:
   (2) 014002  104000                        EMT                     ;BIT DID NOT SET CC'S CORRECTLY
 9187
 9188  014004  000277                BITST2: SCC                     ;CC=1011
 9189  014006  000244                        CLZ
 9190  014010  032700  077776                BIT     #77776,R0       ;CC=0101
 9191  014014  101002                        BHI     BITST3
 9192  014016  102401                        BVS     BITST3
 9193  014020  100001                        BPL     TS211
   (2) 014022                          BITST3:
   (3) 014022  104000                        EMT                     ;BIT DID NOT SET CC'S CORRECTLY
                                      ;*****************************************************************************
 9194                                 ;TEST 211       TEST BIC INSTRUCTION
   (2)                                ;*****************************************************************************
   (3)
   (2) 014024                         TS211:
 9195  014024  012700  177777                MOV     #177777,R0
 9196  014030  000277                        SCC                     ;CC=0110
 9197  014032  000251                        .CLN!CLC
 9198  014034  042700  077777                BIC     #77777,R0       ;CC=1000
 9199  014040  101402                        BLOS    BIC1
 9200  014042  102401                        BVS     BIC1
 9201  014044  100401                        BMI     BIC2
   (1) 014046                          BIC1:
   (2) 014046  104000                        EMT                     ;BIC DID NOT SET CC'S CORRECTLY
 9202  014050  000277                BIC2:   SCC                     ;CC=1011
 9203  014052  000244                        CLZ
 9204  014054  042700  100000                BIC     #100000,R0      ;CC=0101
 9205  014060  101002                        BHI     BIC3
 9206  014062  102401                        BVS     BIC3
 9207  014064  100001                        BPL     TS212
   (2) 014066                          BIC3:
```

```
 (3)  014066  104000                          EMT                     ;BIC DID NOT SET CC'S CORRECTLY
                                      ;*******************************************************************
 (2)                                  ;TEST 212       TEST BIS INSTRUCTION
 (3)                                  ;*******************************************************************
 (2)  014070                          TS212:
      014070  005000                          CLR     R0              ;R0=0
      014072  000277                          SCC                     ;CC=1010
      014074  000251                          +CLN!CLC
      014076  052700  000000                  BIS     #0,R0           ;CC=0100  R0=0
      014102  103403                          BCS     BIS1
      014104  102402                          BVS     BIS1
      014106  100401                          BMI     BIS1
      014110  001401                          BEQ     BIS2
 (1)  014112                          BIS1:
 (2)  014112  104000                          EMT                     ;BIS DID NOT SET CC'S CORRECTLY
      014114  000277                  BIS2:   SCC                     ;CC=0111
      014116  000250                          CLN
      014120  052700  177777                  BIS     #177777,R0      ;CC=1001
      014124  103003                          BCC     BIS3
      014126  102402                          BVS     BIS3
      014130  001401                          BEQ     BIS3
      014132  100401                          BMI     TS213
 (2)  014134                          BIS3:
 (3)  014134  104000                          EMT                     ;BIS DID NOT SET CC'S CORRECTLY

                                      ;*******************************************************************
                                      ;
                                      ;       THESE NEXT TWO TESTS VERIFY THE FUNCTIONING OF THE INC AND
                                      ;DEC INSTRUCTIONS.  THESE INSTRUCTIONS BOTH EFFECT THE C AND V
                                      ;BITS THE SAME; THE C-BIT IS LEFT UNCHANGED AND THE V-BIT IS DEPENDENT
                                      ;UPON THE DATA RESULTS.  THE SAME PROCEDURE IS USED.  THE CONDITION
                                      ;CODE BITS ARE INITIALIZED, THE INSTRUCTION IS EXECUTED AND THE
                                      ;RESULTS ARE VERIFIED WITH A SERIES OF CONDITIONAL BRANCH INSTRUCTIONS.
                                      ;THIS PROCEDURE IS REPEATED WITH SEVERAL DATA PATTERNS TO PRODUCE
                                      ;DIFFERENT COMBINATIONS OF THE C AND V BITS.
                                      ;
                                      ;*******************************************************************
 (2)                                  ;TEST 213       TEST INC INSTRUCTION
 (3)                                  ;*******************************************************************
 (2)  014136                          TS213:
      014136  012700  077777                  MOV     #077777,R0      ;R0=077777
      014142  000257                          CCC                     ;CC=0100
      014144  000264                          SEZ
      014146  005200                          INC     R0              ;CC=1010   R0=10000
      014150  101402                          BLOS    INC1
      014152  100001                          BPL     INC1
      014154  102401                          BVS     INC2
 (1)  014156                          INC1:
 (2)  014156  104000                          EMT                     ;INC DID NOT SET CC'S CORRECTLY
      014160  052700  077777          INC2:   BIS     #77777,R0       ;R0=177777
      014164  000261                          SEC                     ;CC=1011
      014166  000244                          CLZ
      014170  005200                          INC     R0              ;CC=0101   R0=0
      014172  100403                          BMI     INC3
      014174  102402                          BVS     INC3
      014176  103001                          BCC     INC3
```

```
 9251  014200  001401                    BEQ     INC4
  (1)  014202                    INC3:
  (2)  014202  104000                    EMT                     ;INC DID NOT SET CC'S CORRECTLY
 9252
 9253  014204  000277            INC4:   SCC                     ;CC=1110
 9254  014206  000241                    CLC
 9255  014210  005200                    INC     R0              ;CC=0000   R0=1
 9256  014212  101402                    BLOS    INC5
 9257  014214  100401                    BMI     INC5
 9258  014216  100001                    BPL     TS214
  (2)  014220                    INC5:
  (3)  014220  104000                    EMT                     ;INC DID NOT SET CC'S CORRECTLY
 9259
 9260                           ;*****************************************************************************
  (2)                           ;TEST 214       TEST DEC INSTRUCTION
  (3)                           ;*****************************************************************************
  (2)  014222                   TS214:
 9261  014222  012700  000002            MOV     #2,R0           ;R0=2
 9262  014226  000277                    SCC                     ;CC=1111
 9263  014230  005300                    DEC     R0              ;CC=0001   R0=1
 9264  014232  100403                    BMI     DEC1
 9265  014234  001402                    BEQ     DEC1
 9266  014236  102401                    BVS     DEC1
 9267  014240  103401                    BCS     DEC2
  (1)  014242                   DEC1:
  (2)  014242  104000                    EMT                     ;DEC DID NOT SET CC'S CORRECTLY
 9268  014244  000261            DEC2:   SEC                     ;CC=1011
 9269  014246  000244                    CLZ
 9270  014250  005300                    DEC     R0              ;CC=0101   R0=0
 9271  014252  101002                    BHI     DEC3
 9272  014254  100401                    BMI     DEC3
 9273  014256  102001                    BVC     DEC4
  (1)  014260                   DEC3:
  (2)  014260  104000                    EMT                     ;DEC DID NOT SET CC'S CORRECTLY
 9274  014262  000277            DEC4:   SCC                     ;CC=0110
 9275  014264  000251                    .CLN!CLC
 9276  014266  005300                    DEC     R0              ;CC=1000   R0=177777
 9277  014270  101402                    BLOS    DEC5
 9278  014272  102401                    BVS     DEC5
 9279  014274  100401                    BMI     DEC6
  (1)  014276                   DEC5:
  (2)  014276  104000                    EMT                     ;DEC DID NOT SET CC'S CORRECTLY
 9280  014300  042700  077777  DEC6:   BIC     #77777,R0       ;R0=100000
 9281  014304  000277                    SCC                     ;CC=0101
 9282  014306  000252                    .CLN!CLV
 9283  014310  005300                    DEC     R0              ;CC=1011   R0=77777
 9284  014312  100403                    BMI     DEC7            ;CC=0011
 9285  014314  001402                    BEQ     DEC7
 9286  014316  102001                    BVC     DEC7
 9287  014320  103401                    BCS     TS215
  (2)  014322                   DEC7:
  (3)  014322  104000                    EMT                     ;DEC DID NOT SET CC'S CORRECTLY
 9288
 9289
 9290                           ;*****************************************************************************
 9291                           ;
```

```
 9292                                ;       THESE NEXT THREE TESTS VERIFY THE FUNCTIONING OF THE CLR,
 9293                                ;TST, AND SWAB INSTRUCTIONS.  THESE THREE INSTRUCTIONS ALL LEAVE
 9294                                ;THE C AND V BITS CLEARED.  AGAIN, THE CONDITION CODES ARE PRESET,
 9295                                ;THE INSTRUCTION EXECUTED AND THE RESULTS CHECKED WITH CONDITIONAL
 9296                                ;BRANCH INSTRUCTIONS.  THE PROCEDURE IS REPEATED TO PRODUCE OTHER
 9297                                ;COMBINATIONS OF CONDITION CODES.
 9298                                ;
 9299                                ;*****************************************************************************
  (2)                                ;TEST 215       TEST CLR INSTRUCTION
  (3)                                ;*****************************************************************************
  (2)  014324                       TS215:
 9300  014324  000277                        SCC                     ;CC=1011
 9301  014326  000244                        CLZ
 9302  014330  005000                        CLR     R0              ;CC=0100   R0=0
 9303  014332  100403                        BMI     CLR1
 9304  014334  102402                        BVS     CLR1
 9305  014336  103401                        BCS     CLR1
 9306  014340  001401                        BEQ     TS216
  (2)  014342                       CLR1:
  (3)  014342  104000                        EMT                     ;CLR DID NOT SET CC'S CORRECTLY
 9307
 9308                                ;*****************************************************************************
  (2)                                ;TEST 216       TEST TST INSTRUCTION
  (3)                                ;*****************************************************************************
  (2)  014344                       TS216:
 9309  014344  000277                        SCC                     ;CC=1011
 9310  014346  000244                        CLZ
 9311  014350  005700                        TST     R0              ;CC=0100
 9312  014352  100403                        BMI     TEST1
 9313  014354  102402                        BVS     TEST1
 9314  014356  103401                        BCS     TEST1
 9315  014360  001401                        BEQ     TEST2
  (1)  014362                       TEST1:
  (2)  014362  104000                        EMT                     ;TEST DID NOT SET CC'S CORRECTLY
 9316  014364  005300                TEST2:  DEC     R0              ;MAKE R0 NEGATIVE
 9317  014366  000277                        SCC                     ;CC=0111
 9318  014370  000250                        CLN
 9319  014372  005700                        TST     R0              ;CC=1000
 9320  014374  101402                        BLOS    TEST3
 9321  014376  102401                        BVS     TEST3
 9322  014400  100401                        BMI     TS217
  (2)  014402                       TEST3:
  (3)  014402  104000                        EMT                     ;TEST DID NOT SET CC'S CORRECTLY
 9323                                ;*****************************************************************************
  (2)                                ;TEST 217       TEST SWAB INSTRUCTION
  (3)                                ;*****************************************************************************
  (2)  014404                       TS217:
 9324  014404  012700  170000              MOV     #170000,R0      ;R0=170000
 9325  014410  000277                        SCC                     ;CC=0111
 9326  014412  000250                        CLN
 9327  014414  000300                        SWAB    R0              ;CC=1000   R0=360
 9328  014416  101402                        BLOS    SWB1
 9329  014420  102401                        BVS     SWB1
 9330  014422  100401                        BMI     SWB2
  (1)  014424                       SWB1:
  (2)  014424  104000                        EMT                     ;SWAB DID NOT SET CC'S CORRECTLY
```

```
 9331  014426  000277                SWB2:   SCC                     ;CC=1011
 9332  014430  000244                        CLZ
 9333  014432  000300                        SWAB    R0              ;CC=0100   R0=170000
 9334  014434  102403                        BVS     SWB3
 9335  014436  103402                        BCS     SWB3
 9336  014440  100401                        BMI     SWB3
 9337  014442  001401                        BEQ     TS220
   (2) 014444                        SWB3:
   (3) 014444  104000                        EMT                     ;
 9338
 9339                              ;*****************************************************************************
 9340                              ;
 9341                              ;       THESE NEXT TWO TESTS VERIFY THE FUNCTIONING OF THE ADD AND
 9342                              ;ADC INSTRUCTIONS.  BOTH OF THESE INSTRUCTIONS HANDLE THE C AND
 9343                              ;V BITS IDENTICALLY.  THE PROCEDURE IS TO PRESET THE CONDITION
 9344                              ;CODES, EXECUTE THE INSTRUCTION WITH A PARTICULAR SET OF DATA, AND
 9345                              ;THEN CHECK THE RESULTS BY EXECUTING A SERIES OF CONDITIONAL
 9346                              ;BRANCHES.  THES PROCEDURE IS REPEATED SEVERAL TIMES WITH DIFFERENT
 9347                              ;DATA TO PRODUCE EVERY COMBINATION OF C AND V BITS.
 9348                              ;
 9349                              ;*****************************************************************************
   (2)                             ;TEST 220       TEST ADD INSTRUCTION
   (3)                             ;*****************************************************************************
   (2) 014446                      TS220:
 9350  014446  012700  040000                MOV     #40000,R0       ;R0=40000
 9351  014452  000277                        SCC                     ;CC=1111
 9352  014454  062700  030000                ADD     #30000,R0       ;CC=0000   R0=70000
 9353  014460  101402                        BLOS    ADD1
 9354  014462  102401                        BVS     ADD1
 9355  014464  100001                        BPL     ADD2
   (1) 014466                      ADD1:
   (2) 014466  104000                        EMT                     ;ADD DID NOT SET CC'S CORRECTLY
 9356  014470  000264                ADD2:   SEZ                     ;CC=0100
 9357
 9358  014472  062700  010000                ADD     #10000,R0       ;CC=1010   40=100000
 9359  014476  101402                        BLOS    ADD3
 9360  014500  102001                        BVC     ADD3
 9361  014502  100401                        BMI     ADD4
   (1) 014504                      ADD3:
   (2) 014504  104000                        EMT                     ;ADD DID NOT SET CC'S CORRECTLY
 9362  014506  000257                ADD4:   CCC                     ;CC=1000
 9363  014510  000270                        SEN
 9364  014512  062700  100000                ADD     #100000,R0      ;CC=0111   R0=0
 9365  014516  101002                        BMI     ADD5
 9366  014520  102001                        BVC     ADD5
 9367  014522  100001                        BPL     ADD6
   (1) 014524                      ADD5:
   (2) 014524  104000                        EMT                     ;ADD DID NOT SET CC'S CORRECTLY
 9368  014526  062700  177777        ADD6:   ADD     #177777,R0      ;CC=1000   R0=177777
 9369  014532  101402                        BLOS    ADD7
 9370  014534  102401                        BVS     ADD7
 9371  014536  100401                        BMI     ADD8
   (1) 014540                      ADD7:
   (2) 014540  104000                        EMT                     ;ADD DID NOT SET CC'S CORRECTLY
 9372  014542  000277                ADD8:   SCC                     ;CC=1010
 9373  014544  000245                        .CLC!CLZ
```

```
 9374  014546  062700  000001          ADD     #1,R0           ;CC=0101   R=0
 9375  014552  102403                  BVS     ADD9
 9376  014554  103002                  BCC     ADD9
 9377  014556  100401                  BMI     ADD9
 9378  014560  001401                  BEQ     TS221
  (2)  014562                  ADD9:
  (3)  014562  104000                  EMT                     ;ADD DID NOT SET CC'S CORRECTLY
 9379
 9380                          ;*****************************************************************************
  (2)                          ;TEST 221       TEST ADC INSTRUCTION
  (3)                          ;*****************************************************************************
  (2)  014564                  TS221:
 9381  014564  012700  077777          MOV     #077777,R0
 9382  014570  000277                  SCC                     ;CC=0101
 9383  014572  000252                  .CLN!CLV
 9384  014574  005500                  ADC     R0              ;CC=1010
 9385  014576  101402                  BLOS    ADC1
 9386  014600  102001                  BVC     ADC1
 9387  014602  100401                  BMI     ADC2
  (1)  014604                  ADC1:
  (2)  014604  104000                  EMT                     ;ADC DID NOT SET CC'S CORRECTLY
 9388  014606  052700  077777  ADC2:   BIS     #77777,R0
 9389  014612  000277                  SCC                     ;CC=1011
 9390  014614  000244                  CLZ
 9391  014616  005500                  ADC     R0              ;CC=0101   R0=0
 9392  014620  101002                  BHI     ADC3
 9393  014622  102401                  BVS     ADC3
 9394  014624  100001                  BPL     ADC4
  (1)  014626                  ADC3:
  (2)  014626  104000                  EMT                     ;ADC DID NOT SET CC'S CORRECTLY
 9395  014630  000277          ADC4:   SCC
 9396  014632  000245                  .CLZ!CLC                ;CC=1010
 9397  014634  005500                  ADC     R0              ;CC=0100
 9398  014636  102403                  BVS     ADC5
 9399  014640  103402                  BCS     ADC5
 9400  014642  100401                  BMI     ADC5
 9401  014644  001401                  BEQ     TS222
  (2)  014646                  ADC5:
  (3)  014646  104000                  EMT                     ;ADC DID NOT SET CC'S CORRECTLY
 9402
 9403                          ;*****************************************************************************
 9404                          ;
 9405                          ;       THESE NEXT THREE TESTS VERIFY THE FUNCTIONING OF THE NEG,
 9406                          ;CMP, AND COM INSTRUCTIONS.  EACH OF THESE INSTRUCTIONS GENERATE
 9407                          ;THE C AND V BITS IDENTICALLY.  THE CONDITION CODES ARE PRESET,
 9408                          ;THE INSTRUCTIONS EXECUTED, AND THE RESULTS CHECKED WITH A SERIES
 9409                          ;OF CONDITIONAL BRANCH INSTRUCTIONS.  THIS PROCEDURE IS REPEATED
 9410                          ;SEVERAL TIMES WITH DIFFERENT DATA IN ORDER TO GENERATE DIFFERENT
 9411                          ;COMBINATIONS OF THE C AND V BITS.
 9412                          ;
 9413                          ;*****************************************************************************
  (2)                          ;TEST 222       TEST NEG INSTRUCTION
  (3)                          ;*****************************************************************************
  (2)  014650                  TS222:
 9414  014650  012700  000001          MOV     #1,R0
 9415  014654  000277                  SCC                     ;CC=0110
```

```
014656  000251                          +CLN!CLC
014660  005400                          NEG     R0              ;CC=1001   R0=177777
014662  103003                          BCC     NEG1
014664  102402                          BVS     NEG1
014666  001401                          BEQ     NEG1
014670  100401                          BMI     NEG2
014672                          NEG1:
014672  104000                          EMT                     ;NEG DID NOT SET CC'S CORRECTLY
014674  042700  077777          NEG2:   BIC     #77777,R0
014700  000257                          CCC                     ;CC=0100
014702  000264                          SEZ
014704  005400                          NEG     R0              ;CC=1011   R0=100000
014706  102003                          BVC     NEG3
014710  103002                          BCC     NEG3
014712  001401                          BEQ     NEG3
014714  100401                          BMI     NEG4
014716                          NEG3:
014716  104000                          EMT                     ;NEG DID NOT SET CC'S CORRECTLY
014720  005000                  NEG4:   CLR     R0
014722  000277                          SCC                     ;CC=1011
014724  000244                          CLZ
014726  005400                          NEG     R0              ;CC=0100   R0=0
014730  102403                          BVS     NEG5
014732  103402                          BCS     NEG5
014734  001001                          BNE     NEG5
014736  100001                          BPL     TS223
014740                          NEG5:
014740  104000                          EMT                     ;NEG DID NOT SET CC'S CORRECTLY

                                ;*******************************************************************************
                                ;TEST 223       TEST CMP INSTRUCTION
                                ;*******************************************************************************
014742                          TS223:
014742  012700  000005                  MOV     #5,R0
014746  000257                          CCC                     ;CC=1010
014750  000271                          +SEN!SEC
014752  022700  000005                  CMP     #5,R0           ;CC=0101
014756  101002                          BHI     CMP1
014760  102401                          BVS     CMP1
014762  100001                          BPL     CMP2
014764                          CMP1:
014764  104000                          EMT                     ;CMP DID NOT SET CC'S CORRECTLY
014766  012700  100000          CMP2:   MOV     #100000,R0
014772  000277                          SCC                     ;CC=1101
014774  000242                          CLV
014776  020027  077777                  CMP     R0,#77777       ;CC=0010
015002  101402                          BLOS    CMP3
015004  102001                          BVC     CMP3
015006  100001                          BPL     CMP4
015010                          CMP3:
015010  104000                          EMT                     ;CMP DID NOT SET CC'S CORRECTLY
015012  052700  040000          CMP4:   BIS     #40000,R0       ;R0=140000
015016  000257                          CCC                     ;CC=0100
015020  000264                          SEZ
015022  022700  040000                  CMP     #40000,R0       ;CC=1011
015026  102003                          BVC     CMP5
```

```
 9459  015030  103002                BCC     CMP5
 9460  015032  001401                BEQ     CMP5
 9461  015034  100401                BMI     CMP6
                               CMP5:
  (1)  015036
  (2)  015036  104000                EMT                     ;CMP DID NOT SET CC'S CORRECTLY
 9462  015040  042700  040000  CMP6: BIC     #40000,R0
 9463  015044  000277                SCC                     ;CC=1111
 9464  015046  022700  177777        CMP     #-1,R0          ;CC=0000
 9465  015052  101402                BLOS    CMP7
 9466  015054  102401                BVS     CMP7
 9467  015056  100001                BPL     TS224
                               CMP7:
  (2)  015060
  (3)  015060  104000                EMT                     ;CMP DID NOT SET CC'S CORRECTLY
 9468
 9469                          ;*****************************************************************************
  (2)                          ;TEST 224       TEST COM INSTRUCTION
  (3)                          ;*****************************************************************************
  (2)  015062                  TS224:
 9470  015062  012700  177777        MOV     #-1,R0
 9471  015066  000257                CCC                     ;CC=1010
 9472  015070  000265                +SEC!SEZ
 9473  015072  005100                COM     R0              ;CC=0101
 9474  015074  101002                BHI     COM1
 9475  015076  102401                BVS     COM1
 9476  015100  100001                BPL     TS225
                               COM1:
  (2)  015102
  (3)  015102  104000                EMT                     ;COM DID NOT SET CC'S CORRECTLY
 9477
 9478
 9479                          ;*****************************************************************************
 9480                          ;
 9481                          ;       THESE NEXT TWO TESTS VERIFY THE FUNCTIONING OF THE SUB
 9482                          ;AND SBC INSTRUCTIONS.  BOTH OF THESE INSTRUCTIONS HANDLE THE
 9483                          ;C AND V BITS IDENTICALLY.  THE PROCEDURE IS TO PRESET THE CONDITION
 9484                          ;CODES, EXECUTE THE INSTRUCTION WITH A PARTICULAR SET OF DATA, AND
 9485                          ;THEN CHECK THE RESULTS BY EXECUTING A SERIES OF CONDITIONAL
 9486                          ;BRANCHES.  THIS PROCEDURE IS REPEATED SEVERAL TIMES WITH DIFFERENT
 9487                          ;DATA PATTERNS TO PROVIDE EVERY COMBINATION OF THE C AND V BITS.
 9488                          ;
 9489                          ;*****************************************************************************
  (2)                          ;TEST 225       TEST SUB INSTRUCTION
  (3)                          ;*****************************************************************************
  (2)  015104                  TS225:
 9490  015104  012700  125252        MOV     #125252,R0
 9491  015110  000257                CCC                     ;CC=1010
 9492  015112  000271                +SEN!SEC
 9493  015114  162700  125252        SUB     #125252,R0      ;CC=0101   R0=0
 9494  015120  101002                BHI     SUB1
 9495  015122  102401                BVS     SUB1
 9496  015124  100001                BPL     SUB2
                               SUB1:
  (1)  015126
  (2)  015126  104000                EMT                     ;SUB DID NOT SET CC'S CORRECTLY
 9497  015130  052700  100000  SUB2: BIS     #100000,R0
 9498  015134  000277                SCC                     ;CC=1101
 9499  015136  000242                CLV
 9500  015140  162700  077777        SUB     #77777,R0       ;CC=0010   R0=1
```

```
 9501  015144  101402                    BLOS    SUB3
 9502  015146  102001                    BVC     SUB3
 9503  015150  100001                    BPL     SUB4
  (1)  015152                    SUB3:
  (2)  015152  104000                    EMT                     ;
 9504  015154  005100            SUB4:   COM     R0              ;R0=177777
 9505  015156  000277                    SCC                     ;CC=11111
 9506
 9507  015160  162700  100000            SUB     #100000,R0      ;CC=0000   R0=77777
 9508  015164  101402                    BLOS    SUB5
 9509  015166  102401                    BVS     SUB5
 9510  015170  100001                    BPL     SUB6
  (1)  015172                    SUB5:
  (2)  015172  104000                    EMT                     ;SUB DID NOT SET CC'S CORRECTLY
 9511  015174  000257            SUB6:   CCC                     ;CC=0100
 9512  015176  000264                    SEZ
 9513  015200  162700  140000            SUB     #140000,R0      ;CC=1011
 9514  015204  102003                    BVC     SUB7
 9515  015206  103002                    BCC     SUB7
 9516  015210  001401                    BEQ     SUB7
 9517  015212  100401                    BMI     TS226
  (2)  015214                    SUB7:
  (3)  015214  104000                    EMT                     ;
 9518
 9519                           ;*************************************************************************************
  (2)                           ;TEST 226       TEST SBC INSTRUCTION
  (3)                           ;*************************************************************************************
  (2)  015216                   TS226:
 9520  015216  012700  000001            MOV     #1,R0
 9521  015222  000277                    SCC                     ;CC=1011
 9522  015224  000244                    CLZ
 9523  015226  005600                    SBC     R0              ;CC=0100   R=0
 9524  015230  103403                    BCS     SBC1
 9525  015232  102402                    BVS     SBC1
 9526  015234  100401                    BMI     SBC1
 9527  015236  001401                    BEQ     SBC2
  (1)  015240                   SBC1:
  (2)  015240  104000                    EMT                     ;SBC DID NOT SET CC'S CORRECTLY
 9528  015242  000277            SBC2:   SCC                     ;CC=1010
 9529  015244  000245                    +CLZ!CLC
 9530  015246  005600                    SBC     R0              ;CC=0100   R=0
 9531  015250  103403                    BCS     SBC3
 9532  015252  102402                    BVS     SBC3
 9533  015254  100401                    BMI     SBC3
 9534  015256  001401                    BEQ     SBC4
  (1)  015260                   SBC3:
  (2)  015260  104000                    EMT                     ;SBC DID NOT SET CC'S CORRECTLY
 9535  015262  000277            SBC4:   SCC                     ;CC=0111
 9536  015264  000250                    CLN
 9537  015266  005600                    SBC     R0              ;CC=1001   R0=177777
 9538  015270  103003                    BCC     SBC5
 9539  015272  102402                    BVS     SBC5
 9540  015274  001401                    BEQ     SBC5
 9541  015276  100401                    BMI     SBC6
  (1)  015300                   SBC5:
  (2)  015300  104000                    EMT                     ;SBC DID NOT SET CC'S CORRECTLY
```

```
 9542  015302  042700  077777          SBC6:   BIC     #77777,R0       ;R0=100000
 9543  015306  000277                          SCC                     ;CC=1101
 9544  015310  000242                          CLV
 9545  015312  005600                          SBC     R0              ;CC=0010
 9546  015314  101402                          BLOS    SBC7
 9547  015316  102001                          BVC     SBC7
 9548  015320  100001                          BPL     TS227
  (2)  015322                          SBC7:
  (3)  015322  104000                          EMT                     ;SBC DID NOT SET CC'S CORRECTLY
 9549
 9550                                 ;*****************************************************************************
 9551                                 ;
 9552                                 ;       THESE NEXT FOUR TESTS VERIFY THE FUNCTIONING OF THE ROL,
 9553                                 ;ROR, ASL AND ASR INSTRUCTIONS.  SPECIAL DATA PATTERNS ARE LOADED
 9554                                 ;AND ROTATED SEVERAL TIMES FOR EACH TEST.  THE CONDITION CODES
 9555                                 ;ARE PRESET BEFORE EACH ROTATION AND THE CONDITION CODES ARE
 9556                                 ;CHECKED AFTER EACH ROTATION.  THE FINAL CHECK IN EACH TEST IS
 9557                                 ;TO VERIFY THE COMMULATIVE DATA RESULT.  THE DATA PATTERNS HAVE
 9558                                 ;BEEN SELECTED TO PRODUCE ALL COMBINATIONS OF THE C AND V BITS.
 9559                                 ;
 9560                                 ;*****************************************************************************
  (2)                                 ;TEST 227       TEST ROL INSTRUCTION
  (3)                                 ;*****************************************************************************
  (2)  015324                         TS227:
 9561  015324  012700  144000                MOV     #144000,R0      ;R0=144000
 9562  015330  000257                        CCC                     ;CC=0110
 9563  015332  000266                        .SEZ!SEV
 9564  015334  006100                        ROL     R0              ;CC=1001   R0=110000
 9565  015336  103003                        BCC     ROL1
 9566  015340  102402                        BVS     ROL1
 9567  015342  001401                        BEQ     ROL1
 9568  015344  100401                        BMI     ROL2
  (1)  015346                         ROL1:
  (2)  015346  104000                        EMT                     ;
 9569  015350  000277                ROL2:   SCC                     ;CC=1100
 9570  015352  000243                        .CLV!CLC
 9571  015354  006100                        ROL     R0              ;CC=0011   R0=020000
 9572  015356  103003                        BCC     ROL3
 9573  015360  102002                        BVC     ROL3
 9574  015362  001401                        BEQ     ROL3
 9575  015364  100001                        BPL     ROL4
  (1)  015366                         ROL3:
  (2)  015366  104000                        EMT                     ;ROL DID NOT SET CC'S CORRECTLY
 9576  015370  000277                ROL4:   SCC                     ;CC=0111
 9577  015372  000250                        CLN
 9578  015374  006100                        ROL     R0              ;CC=0000   R0=040001
 9579  015376  101402                        BLOS    ROL5
 9580  015400  102401                        BVS     ROL5
 9581  015402  100001                        BPL     ROL6
  (1)  015404                         ROL5:
  (2)  015404  104000                        EMT                     ;ROL DID NOT SET CC'S CORRECTLY
 9582  015406  000257                ROL6:   CCC                     ;CC=0101
 9583  015410  000265                        .SEZ!SEC
 9584  015412  006100                        ROL     R0              ;CC=1010   R0=100003
 9585  015414  101405                        BLOS    ROL7
 9586  015416  102004                        BVC     ROL7
```

```
 9587  015420  100003                        BPL     ROL7
 9588  015422  022700  100003                CMP     #100003,R0
 9589  015426  001401                        BEQ     TS230
  (2)  015430                          ROL7:
  (3)  015430  104000                        EMT                     ;ROL MALFUNCTIONED
 9590
  (2)                                 ;*****************************************************************************
  (3)                                 ;TEST 230       TEST ROR INSTRUCTION
  (2)  015432                         ;*****************************************************************************
                                      TS230:
 9591  015432  012700  000023                MOV     #23,R0          ;R0=23
 9592  015436  000277                        SCC                     ;CC=0111
 9593  015440  000250                        CLN
 9594  015442  006000                        ROR     R0              ;CC=1001   R0=100011
 9595  015444  102403                        BVS     ROR1
 9596  015446  103002                        BCC     ROR1
 9597  015450  001401                        BEQ     ROR1
 9598  015452  100401                        BMI     ROR2
  (1)  015454                          ROR1:
  (2)  015454  104000                        EMT                     ;ROR DID NOT SET CC'S CORRECTLY
 9599  015456  000257                  ROR2: CCC                     ;CC=1100
 9600  015460  000274                        .SEN!SEZ
 9601  015462  006000                        ROR     R0              ;CC=0011   R0=040004
 9602  015464  102003                        BVC     ROR3
 9603  015466  103002                        BCC     ROR3
 9604  015470  001401                        BEQ     ROR3
 9605  015472  100001                        BPL     ROR4
  (1)  015474                          ROR3:
  (2)  015474  104000                        EMT                     ;ROR DID NOT SET CC'S CORRECTLY
 9606  015476  000277                  ROR4: SCC                     ;CC=1110
 9607  015500  000241                        CLC
 9608  015502  006000                        ROR     R0              ;CC=0000   R0=020002
 9609  015504  101403                        BLOS    ROR5
 9610  015506  102402                        BVS     ROR5
 9611  015510  001401                        BEQ     ROR5
 9612  015512  100001                        BPL     ROR6
  (1)  015514                          ROR5:
  (2)  015514  104000                        EMT                     ;ROR DID NOT SET CC'S CORRECTLY
 9613  015516  000257                  ROR6: CCC                     ;CC=0101
 9614  015520  000265                        .SEC!SEZ
 9615  015522  006000                        ROR     R0              ;CC=1010   R0=110001
 9616  015524  101402                        BLOS    ROR7
 9617  015526  102001                        BVC     ROR7
 9618  015530  100401                        BMI     TS231
  (2)  015532                          ROR7:
  (3)  015532  104000                        EMT                     ;ROR DID NOT PRODUCE CORRECT RESULTS
 9619
  (2)                                 ;*****************************************************************************
  (3)                                 ;TEST 231       TEST ASL INSTRUCTION
  (2)  015534                         ;*****************************************************************************
                                      TS231:
 9620  015534  012700  144000                MOV     #144000,R0      ;R0=14000
 9621  015540  000257                        CCC                     ;CC=0110
 9622  015542  000271                        .SEN!SEC
 9623  015544  006300                        ASL     R0              ;CC=1001   R0=110000
 9624  015546  103003                        BCC     ASL1
 9625  015550  102402                        BVS     ASL1
 9626  015552  001401                        BEQ     ASL1
```

```
 9627  015554  100401                        BMI     ASL2
  (1)  015556                          ASL1:
  (2)  015556  104000                        EMT                     ;
 9628  015560  000277                  ASL2: SCC                     ;CC=1100
 9629  015562  000243                        .CLV!CLC
 9630  015564  006300                        ASL     R0              ;CC=0011   R0=020000
 9631  015566  103003                        BCC     ASL3
 9632  015570  102002                        BVC     ASL3
 9633  015572  001401                        BEQ     ASL3
 9634  015574  100001                        BPL     ASL4
  (1)  015576                          ASL3:
  (2)  015576  104000                        EMT                     ;ASL DID NOT SET CC'S CORRECTLY
 9635  015600  000277                  ASL4: SCC                     ;CC=0111
 9636  015602  000250                        CLN
 9637  015604  006300                        ASL     R0              ;CC=0000   R0=040000
 9638  015606  101402                        BLOS    ASL5
 9639  015610  102401                        BVS     ASL5
 9640  015612  100001                        BPL     ASL6
  (1)  015614                          ASL5:
  (2)  015614  104000                        EMT                     ;ASL DID NOT SET CC'S CORRECTLY
 9641  015616  000257                  ASL6: CCC                     ;CC=0101
 9642  015620  000265                        .SEZ!SEC
 9643  015622  006300                        ASL     R0              ;CC=1010   R0=100000
 9644  015624  103406                        BCS     ASL7
 9645  015626  001405                        BEQ     ASL7
 9646  015630  102004                        BVC     ASL7
 9647  015632  100003                        BPL     ASL7
 9648  015634  022700  100000                CMP     #100000,R0
 9649  015640  001401                        BEQ     TS232
  (2)  015642                          ASL7:
  (3)  015642  104000                        EMT                     ;ASL MALFUNCTIONED
 9650                                ;*******************************************************************************
  (2)                                ;TEST 232       TEST ASR INSTRUCTION
  (3)                                ;*******************************************************************************
  (2)  015644                        TS232:
 9651  015644  012700  100023                MOV     #100023,R0      ;R0=100023
 9652  015650  000277                        SCC                     ;CC=0110
 9653  015652  000250                        CLN
 9654  015654  006200                        ASR     R0              ;CC=1001   RP=140011
 9655  015656  102403                        BVS     ASR1
 9656  015660  103002                        BCC     ASR1
 9657  015662  001401                        BEQ     ASR1
 9658  015664  100401                        BMI     ASR2
  (1)  015666                          ASR1:
  (2)  015666  104000                        EMT                     ;ASR DID NOT SET CC'S CORRECTLY
 9659  015670  042700  100000          ASR2: BIC     #100000,R0      ;R0=40011
 9660  015674  000277                        SCC                     ;CC=1100
 9661  015676  000243                        .CLV!CLC
 9662  015700  006200                        ASR     R0              ;CC=0011   R0=020004
 9663  015702  102003                        BVC     ASR3
 9664  015704  103002                        BCC     ASR3
 9665  015706  001401                        BEQ     ASR3
 9666  015710  100001                        BPL     ASR4
  (1)  015712                          ASR3:
  (2)  015712  104000                        EMT                     ;ASR DID NOT SET CC'S CORRECTLY
 9667  015714  000277                  ASR4: SCC                     ;CC=1111
```

```
 9668
 9669  015716  006200                        ASR     R0              ;CC=0000   R0=010002
 9670  015720  101403                        BLOS    ASR5
 9671  015722  102402                        BVS     ASR5
 9672  015724  001401                        BEQ     ASR5
 9673  015726  100001                        BPL     ASR6
  (1)  015730                          ASR5:
  (2)  015730  104000                        EMT                     ;ASR DID NOT SET CC'S CORRECTLY
 9674  015732  052700  100000          ASR6: BIS     #100000,R0      ;R0=110002
 9675  015736  000257                        CCC                     ;CC=0101
 9676  015740  000265                        .SEZ!SEC
 9677  015742  006200                        ASR     R0              ;C=1010   R0=144001
 9678  015744  101406                        BLOS    ASR7
 9679  015746  102005                        BVC     ASR7
 9680  015750  100004                        BPL     ASR7
 9681  015752  001403                        BEQ     ASR7
 9682  015754  022700  144001              CMP     #144001,R0      ;CHECK RESULT OF ASR'S
 9683  015760  001401                        BEQ     TS233
  (2)  015762                          ASR7:
  (3)  015762  104000                        EMT                     ;ASR DID NOT FUNCTION CORRECTLY
 9684
 9685                                ;*****************************************************************************
  (2)                                ;TEST 233       TEST RORB INSTRUCTION
  (3)                                ;*****************************************************************************
  (2)  015764                        TS233:
 9687  015764  112701  000004              MOVB    #4,R1   ;LOAD REGISTER
 9688  015770  000257                      CCC                     ;CLEAR ALL FLAGS
 9689  015772  106001                      RORB    R1              ;SHIFT  BYTE RIGHT
 9690  015774  106001                      RORB    R1              ;SHIFT BYTE RIGHT
 9691  015776  122701  000001              CMPB    #1,R1           ;CHECK RESULT
 9692  016002  001401                      BEQ     RORB1
  (2)  016004  104000                      EMT                     ;RORB DID NOT FUNCTION CORRECTLY
 9693  016006  106001                RORB1: RORB    R1              ;SHIFT BYTE RIGHT
 9694  016010  100403                      BMI     RORB2           ;CC=7?
 9695  016012  001002                      BNE     RORB2
 9696  016014  102001                      BVC     RORB2
 9697  016016  103401                      BCS     RORB3
  (1)  016020                          RORB2:
  (2)  016020  104000                      EMT                     ;RORB DID NOT SET CC'S CORRECTLY
 9698  016022  106001                RORB3: RORB    R1              ;SHIFT BYTE RIGHT
 9699  016024  100002                      BPL     RORB4           ;CC=12
 9700  016026  101401                      BLOS    RORB4
 9701  016030  102401                      BVS     RORB5
  (1)  016032                          RORB4:
  (2)  016032  104000                      EMT                     ;RORB DID NOT SET CC CORRECTLY
 9702  016034  122701  000200          RORB5: CMPB    #200,R1         ;CHECK RESULT
 9703  016040  001401                      BEQ     RORB7
  (2)  016042  104000                      EMT                     ;RORB DID NOT FUNCTION CORRECTLY
 9704  016044                         RORB7:
 9705                                ;ROTATE ODD BYTE
 9706  016044  005000                      CLR     R0              ;MAKE R0 ZERO
 9707  016046  012710  025125              MOV     #025125,(R0)    ;PUT STARTING VALUE IN LOC. 0
 9708  016052  005200                      INC     R0              ;MAKE R0 POINT TO ODD BYTE
 9709  016054  000257                      CCC                     ;CLEAR ALL CC
 9710  016056  000261                      SEC                     ;SEC CARRY BIT
```

F8



```
016060  106010                          RORB    (R0)            ;SHIFT BYTE RIGHT
016062  100002                          BPL     RORB10          ;CC=12?
016064  101401                          BLOS    RORB10
016066  102401                          BVS     RORB11
016070                          RORB10:
016070  104000                          EMT                     ;RORB DID NOT SET CC'S CORRECTLY
016072  022737  112525  000000  RORB11: CMP     #112525,@#0     ;CHECK RESULT
016100  001401                          BEQ     RORB12
016102  104000                          EMT                     ;RORB DID NOT FUNCTION CORRECTLY
016104  106010                  RORB12: RORB    (R0)            ;SHIFT BYTE RIGHT
016106  100403                          BMI     RORB13          ;CC=3?
016110  001402                          BEQ     RORB13
016112  102001                          BVC     RORB13
016114  103401                          BCS     RORB14
016116                          RORB13:
016116  104000                          EMT                     ;RORB DID NOT SET CC CORRECTLY
016120  022737  045125  000000  RORB14: CMP     #045125,@#0     ;CHECK RESULT
016126  001401                          BEQ     TS234
016130  104000                          EMT                     ;RORB DID NOT FUNCTION CORRECTLY


                                ;*******************************************************************************
                                ;TEST 234       TEST ASLB INSTRUCTION
                                ;*******************************************************************************
016132                          TS234:
016132  112701  000040                  MOVB    #40,R1          ;LOAD REGISTER
016136  000257                          CCC                     ;CLEAR ALL CONDITION CODES
016140  106301                          ASLB    R1              ;SHIFT BYTE LEFT
016142  106301                          ASLB    R1              ;SHIFT BYTE LEFT
016144  100002                          BPL     ASLB2           ;CHECK CC=12
016146  101401                          BLOS    ASLB2
016150  102401                          BVS     ASLB3
016152                          ASLB2:
016152  104000                          EMT                     ;ASLB DID NOT SET CONDITION CODE CORRECTLY
016154  022701  000200          ASLB3:  CMP     #200,R1         ;CHECK RESULT
016160  001401                          BEQ     ASLB1
016162  104000                          EMT                     ;ASLB DID NOT FUNCTION CORRECTLY
016164  106301                  ASLB1:  ASLB    R1              ;SHIFT BYTE LEFT
016166  100403                          BMI     ASLB4           ;CHECK CC=7?
016170  001002                          BNE     ASLB4
016172  102001                          BVC     ASLB4
016174  103401                          BCS     TS235
016176                          ASLB4:
016176  104000                          EMT                     ;ASLB   DID NOT SET CC'S CORRECTLY


                                ;*******************************************************************************
                                ;TEST 235       TEST ASRB INSTRUCTION
                                ;*******************************************************************************
016200                          TS235:
016200  112701  000004                  MOVB    #4,R1           ;SET UP STARTING DATA
016204  000257                          CCC                     ;CLEAR ALL CONDITION CODES
016206  106201                          ASRB    R1              ;SKIFT BYTE RIGHT
016210  106201                          ASRB    R1              ;SHIFT BYTE RIGHT
016212  122701  000001                  CMPB    #1,R1           ;CHECK DATA
016216  001401                          BEQ     ASRB1
```

```
  (2)  016220  104000                    EMT                     ;ASRB DID NOT SHIFT DATA CORRECTLY
9750   016222  106201          ASRB1:    ASRB    R1              ;SHIFT BYTE RIGHT
9751   016224  100403                    BMI     ASRB2           ;CHECK CONDITION CODE = 7?
9752   016226  001002                    BNE     ASRB2
9753   016230  102001                    BVC     ASRB2
9754   016232  103401                    BCS     ASRB3
  (1)  016234                    ASRB2:
  (2)  016234  104000                    EMT                     ;ASRB DID NOT SET CC'S CORRECTLY
9755   016236  106201          ASRB3:    ASRB    R1              ;SHIFT BYTE RIGHT
9756   016240  103401                    BCS     ASRB4           ;CHECK CC=4
9757   016242  001401                    BEQ     ASRB5
  (1)  016244                    ASRB4:
  (2)  016244  104000                    EMT                     ;ASRB DID NOT SET CC'S CORRECTLY
9758   016246  112701  000202    ASRB5:  MOVB    #202,R1         ;PUT STARTING DATA IN REGISTER
9759   016252  106201                    ASRB    R1              ;SHIFT BYTE RIGHT
9760   016254  106201                    ASRB    R1              ;SHIFT BYTE RIGHT
9761   016256  100003                    BPL     ASRB6           ;CHECK CC'S =11?
9762   016260  001402                    BEQ     ASRB6
9763   016262  102401                    BVS     ASRB6
9764   016264  103401                    BCS     ASRB7
  (1)  016266                    ASRB6:
  (2)  016266  104000                    EMT                     ;ASRB DID NOT SET CC'S CORRECTLY
9765   016270  122701  000340    ASRB7:  CMPB    #340,R1         ;CHECK RESULT
9766   016274  001401                    BEQ     TS236
  (3)  016276  104000                    EMT                     ;ASRB DID NOT SHIFT DATA CORRECTLY
9767
9768                            ;*****************************************************************************
9769                            ;
9770                            ;       THIS TEST VERIFIES THE SXT INSTRUCTION. CONDITION  CODES
9771                            ;ARE PRESET IN EACH OF THE TWO POSSIBLE CASES. WITH THE N-BIT SET,
9772                            ;THE TEST CHECKS FOR ALL ONES IN THE DESTINATION. WITH THE N-BIT
9773                            ;CLEAR, THE DESTINATION SHOULD CONTAIN ALL ZEROES. THE DATA
9774                            ;IS VERIFIED BY CONDITIONAL BRANCHES.
9775                            ;
9776                            ;*****************************************************************************
 (2)                            ;TEST 236       TEST THE SXT INSTRUCTION
 (3)                            ;*****************************************************************************
 (2)  016300                    TS236:
9777   016300  005000                    CLR     R0
9778   016302  000277                    SCC                     ;SET CC=1011
9779   016304  000244                    CLZ
9780   016306  006700                    SXT     R0              ;TRY SXT
9781   016310  100006                    BPL     SXT0            ;TEST CC=1001
9782   016312  001405                    BEQ     SXT0
9783   016314  102404                    BVS     SXT0
9784   016316  103003                    BCC     SXT0
9785   016320  022700  177777          CMP     #-1,R0          ;CHECK DATA RESULT
9786   016324  001401                    BEQ     SXT1
  (1)  016326                    SXT0:
  (2)  016326  104000                    EMT                     ;RESULTS OF SXT INCORRECT
9787   016330  005000          SXT1:     CLR     R0              ;R0=0
9788   016332  005010                    CLR     (R0)            ;LOC. 0=0
9789   016334  005110                    COM     (R0)            ;LOC. 0=177777
9790   016336  000257                    CCC                     ;SET CC=0110
9791   016340  000266                    .SEZ!SEV
9792   016342  006710                    SXT     (R0)
```

```
 9793  016344  001005                    BNE     SXT2            ;TEST CC=0100
 9794  016346  103404                    BCS     SXT2
 9795  016350  102403                    BVS     SXT2
 9796  016352  100402                    BMI     SXT2
 9797  016354  005710                    TST     (R0)
 9798  016356  001401                    BEQ     TS237
  (2)  016360                        SXT2:
  (3)  016360  104000                    EMT                     ;RESULTS OF SXT INCORRECT
 9799                                ;*****************************************************************************
 9800                                ;
```

```
                                ;       THIS TEST VERIFIES THE XOR INSTRUCTION. UNIQUE PATTERNS
                                ;OF ONES AND ZEROES ARE MOVED TO DATA REGISTERS R0 AND R1.
                                ;AFTER THE FIRST XOR INSTRUCTION R0=36146. AN XOR IS THEN
                                ;EXECUTED WITH THIS NEW VALUE AND THE CONTENTS OF R1 TO
                                ;REPRODUCE THE ORIGINAL VALUE IF R0=31525.
                                ;
                                ;*****************************************************************************************
                                ;TEST 237       TEST THE XOR INSTRUCTION
                                ;*****************************************************************************************
                                TS237:
 9809  016362  012700  007463          MOV     #7463,R0        ;SET UP R0
 9810  016366  012701  031525          MOV     #31525,R1       ;SET UP R1
 9811  016372  000277                  SCC                     ;SET CC=1110
 9812  016374  000241                  CLC
 9813  016376  074100                  XOR     R1,R0           ;TRY XOR
 9814  016400  101406                  BLOS    XOR1            ;CC=0000?
 9815  016402  102405                  BVS     XOR1
 9816  016404  001404                  BEQ     XOR1
 9817  016406  100403                  BMI     XOR1
 9818  016410  022700  036146          CMP     #36146,R0       ;DATA RESULT CORRECT?
 9819  016414  001401                  BEQ     XOR2
                                XOR1:
 (1)   016416
 (2)   016416  104000                  EMT                     ;
                                XOR2:
 9820  016420  010104                  MOV     R1,R4
 9821  016422  000261                  SEC                     ;CC=1110
 9822  016424  000241                  CLC
 9823  016426  074400                  XOR     R4,R0           ;TRY XOR MODE 0,0
 9824  016430  101406                  BLOS    XOR3            ;CC=0000?
 9825  016432  102405                  BVS     XOR3
 9826  016434  001404                  BEQ     XOR3
 9827  016436  100403                  BMI     XOR3
 9828  016440  022700  007463          CMP     #7463,R0
 9829  016444  001401                  BEQ     TS240
                                XOR3:
 (2)   016446
 (3)   016446  104000                  EMT                     ;RESULT OF XOR INCORRECT
                                ;*****************************************************************************************
                                ;
                                ;       THIS TEST VERIFIES THE SOB INSTRUCTION. R4 IS USED AS A
                                ;COUNTER WHILE R0 IS THE ADDRESS REGISTER. CONDITIONAL
                                ;BRANCHES ARE USED TO VERIFY PROPER TRANSFER OF CONTROL
                                ;WHILE R4 IS CHECKED TO INSURE PROPER DECREMENTING OF R0.
                                ;
                                ;*****************************************************************************************
                                ;TEST 240       TEST SOB INSTRUCTION
                                ;*****************************************************************************************
                                TS240:
 9838  016450  012700  000525          MOV     #525,R0
 9839  016454  010004                  MOV     R0,R4
 9840  016456  000277                  SCC                     ;SET CC=1111
                                SOB1:   BHI     SOB2            ;CC=1111?
 9842  016462  100001                  BPL     SOB2
 9843  016464  102401                  BVS     SOB3
                                SOB2:
 (1)   016466
 (2)   016466  104000                  EMT                     ;
                                SOB3:   DEC     R4              ;COUNT ITERATIONS
 9845  016472  000277                  SCC                     ;CC=1111
```

```
 9846  016474  077007                 SOB     R0,SOB1         ;DO SOB W/ R0
 9847  016476  101004                 BHI     SOB4            ;CHECK CC=1111
 9848  016500  100003                 BPL     SOB4
 9849  016502  102002                 BVC     SOB4
 9850  016504  005704                 TST     R4              ;ITERATION COUNT OK?
 9851  016506  001401                 BEQ     TS241
  (2)  016510                  SOB4:
  (3)  016510  104000                 EMT                     ;INCORRECT # OF BRANCHES OR CC'S CHANGED
 9852                          ;*****************************************************************************
 9853                          ;
 9854                          ;       THIS TEST VERIFIES THE MARK INSTRUCTION. THE EFFECTS
 9855                          ;OF THE MARK INSTRUCTION ARE SIMULATED BY THE PROGRAM INSTRUCTIONS.
 9856                          ;THE CONTENTS OF R5 AND THE STACK POINTER ARE CHECKED AFTER EACH
 9857                          ;OF THE TWO ROUTINES IN THE TEST.
 9858                          ;
 9859                          ;*****************************************************************************
  (2)                          ;TEST 241       TEST MARK INSTRUCTION
  (3)                          ;*****************************************************************************
  (2)  016512                  TS241:
 9860  016512  012706  001000         MOV     #STBOT,SP
 9861  016516  012746  125252         MOV     #125252,-(SP)   ;PUT R5 VALUE ON STACK
 9862  016522  162706  000074         SUB     #74,SP          ;EFFECTIVELY PUT 36 ARGUMENTS ON STACK
 9863  016526  012705  016544         MOV     #MRK1,R5        ;SET NEW PC IN R5
 9864  016532  012746  006436         MOV     #6436,-(SP)     ;PUT MARK 36 INST. ON STACK
 9865  016536  000277                 SCC                     ;SET CC=1111
 9866  016540  000116                 JMP     (SP)            ;XFER CONTL TO MARK 36 INST. ON STACK
 9867  016542  104000                 EMT                     ;MARK INST. SHOULD HAVE JUMPED TO MRK1
 9868  016544  101010          MRK1:  BHI     MRK2            ;TEST CC UNAFFECTED
 9869  016546  100007                 BPL     MRK2            ;IE. CC=1111
 9870  016550  102006                 BVC     MRK2
 9871  016552  020527  125252         CMP     R5,#125252      ;CHECK R5 RESTORED FROM STACK
 9872  016556  001003                 BNE     MRK2
 9873  016560  022706  001000         CMP     #STBOT,R6       ;CHECK STACK POINTER READJUSTED CORRECTLY.
 9874  016564  001401                 BEQ     MRK3
  (1)  016566                  MRK2:
  (2)  016566  104000                 EMT                     ;RESULTS OF MARK INCORRECT
 9875  016570  012746  052525  MRK3:  MOV     #52525,-(SP)
 9876  016574  012746  006400         MOV     #6400,-(SP)     ;PUT MARK 0 INST. ON STACK
 9877  016600  010605                 MOV     SP,R5           ;SET ADDR. OF MARK INST. IN R5
 9878  016602  004737  016612         JSR     PC,@#MRK4       ;DO JSR
 9879  016606  000137  016616         JMP     @#MRK5          ;
 9880  016612  000205          MRK4:  RTS     R5              ;DO RTS WITH R5 TO MARK INST ON STACK
 9881  016614  104000                 EMT                     ;RTS,MARK SEQUENCE FAILED
 9882  016616  022706  001000  MRK5:  CMP     #STBOT,R6       ;STACK ADJUSTED CORRECTLY
 9883  016622  001003                 BNE     MRK6            ;IF NOT:  BR
 9884  016624  022705  052525         CMP     #52525,R5       ;CHECK IF R5 RESTORED FROM STACK
 9885  016630  001401                 BEQ     TS242
  (2)  016632                  MRK6:
  (3)  016632  104000                 EMT                     ;RESULTS OF MARK INCORRECT
 9886          177776                 PS=177776
 9887                          ;*****************************************************************************
 9888                          ;
 9889                          ;       THESE NEXT SEVEN TESTS VERIFY THE MTPS INSTRUCTION IN ALL
 9890                          ;       MODES. THE PSW IS DEFINED BY AN EQUATE STATEMENT BEFORE THE
 9891                          ;       FIRST MTPS TEST. IN EACH TEST A PATTERN OF ONES AND
 9892                          ;       ZEROES IS SET IN A DATA REGISTER AND MOVED TO THE PSW.
```

```
 9893                                   ;       THE DATA IN THE PSW, AND THE DATA REGISTER ADDRESS,
 9894                                   ;       ARE CHECKED TO VERIFY PROPER EXECUTION OF THE INSTRUCTION.
 9895                                   ;
 9896                                   ;*****************************************************************************************
  (2)                                   ;TEST 242       TEST MTPS INSTRUCTION
  (3)                                   ;*****************************************************************************************
  (2)  016634                           TS242:
 9897  016634  012700  000377                        MOV     #377,R0
 9898  016640  000257                                CCC
 9899  016642  106400                                MTPS    R0
 9900  016644  022767  000357  161124                CMP     #357,PS
 9901  016652  001401                                BEQ     MTPS1
  (2)  016654  104000                                EMT                             ;MTPS FAILED
 9902  016656  005000                        MTPS1:  CLR     R0
 9903  016660  005010                                CLR     (R0)
 9904  016662  000277                                SCC                             ;CC=1111
 9905  016664  106410                                MTPS    (R0)                    ;TRY MTPS MODE 1
 9906  016666  100403                                BMI     MTPS1A                  ;CHECK PS
 9907  016670  102402                                BVS     MTPS1A
 9908  016672  103401                                BCS     MTPS1A
 9909  016674  001001                                BNE     TS243
  (2)  016676                           MTPS1A:
  (3)  016676  104000                                EMT                             ;MTPS FAILED
 9910                                   ;*****************************************************************************************
 9911                                   ;TEST 243       TEST MTPS MODE 2
  (2)                                   ;*****************************************************************************************
  (3)                                   TS243:
  (2)  016700
 9912  016700  005000                        CLR     R0                      ;R0=0
 9913  016702  012710  177777                MOV     #-1,(R0)                ;LOC. 0=-1
 9914  016706  005037  177776                CLR     @#PS                    ;PS=0
 9915  016712  106420                        MTPS    (R0)+                   ;TRY MTPS W/MODE 2
 9916  016714  022737  000357  177776        CMP     #357,@#PS               ;CHECK DATA
 9917  016722  001401                        BEQ     MTPS2
  (2)  016724  104000                        EMT                             ;DEST. DATA INCORRECT
 9918  016726  022700  000001        MTPS2:  CMP     #1,R0                   ;CHECK DEST. REGISTER.
 9919  016732  001401                        BEQ     TS244
  (3)  016734  104000                        EMT                             ;DEST REGISTER NOT INCREMENTED BY 1
 9920
 9921                                   ;*****************************************************************************************
  (2)                                   ;TEST 244       TEST MTPS MODE 3
  (3)                                   ;*****************************************************************************************
  (2)  016736                           TS244:
 9922  016736  012700  000402                MOV     #402,R0                 ;R0=402
 9923  016742  005010                        CLR     (R0)                    ;LOC. 402=0
 9924  016744  012737  052652  000000        MOV     #52652,@#0              ;LOC. 0=52652
 9925  016752  005037  177776                CLR     @#PS                    ;PS=0
 9926  016756  106430                        MTPS    @(R0)+                  ;TRY MTPS W/MODE 3
 9927  016760  022737  000252  177776        CMP     #252,@#PS               ;CHECK DEST. DATA
 9928  016766  001401                        BEQ     MTPS3
  (2)  016770  104000                        EMT                             ;DEST. DATA INCORRECT
 9929  016772  022700  000404        MTPS3:  CMP     #404,R0                 ;CHECK MODE 3 REGISTER.
 9930  016776  001401                        BEQ     TS245
  (3)  017000  104000                        EMT                             ;MODE 3 REGISTER INCORRECT
 9931
 9932                                   ;*****************************************************************************************
```

```
 (2)                                    ;TEST 245       TEST MTPS MODE 4
 (3)                                    ;*****************************************************************************
 (2)  017002                            TS245:
9933  017002  012700  000001                    MOV     #1,R0           ;R0=1
9934  017006  012737  125125  000000            MOV     #125125,@#0     ;LOC. 0 = 125125
9935  017014  005037  177776                    CLR     @#PS            ;PS=0
9936  017020  106440                            MTPS    -(R0)           ;TRY MTPS W/MODE 4
9937  017022  022737  000105  177776            CMP     #105,@#PS       ;CHECK DEST. DATA
9938  017030  001401                            BEQ     MTPS4
 (2)  017032  104000                            EMT                     ;DEST. DATA INCORRECT
9939  017034  005700                    MTPS4:  TST     R0              ;CHECK MODE 4 REGISTER
9940  017036  001401                            BEQ     TS246
 (3)  017040  104000                            EMT                     ;MODE 4 REGISTER NOT DECREMENTED BY 1
9941
9942                                    ;*****************************************************************************
 (2)                                    ;TEST 246       TEST MTPS MODE 5
 (3)                                    ;*****************************************************************************
 (2)  017042                            TS246:
9943  017042  012700  000404                    MOV     #404,R0         ;R0=404
9944  017046  012737  177400  000000            MOV     #177400,@#0     ;LOC. 0=177400
9945  017054  000277                            SCC                     ;SET ALL COND. CODES
9946  017056  106450                            MTPS    @-(R0)          ;TRY MTPS W/MODE 5
9947  017060  005737  177776                    TST     @#PS            ;CHECK DEST. DATA.
9948  017064  001401                            BEQ     MTPS5
 (2)  017066  104000                            EMT                     ;DESTINATION DATA INCORRECT
9949  017070  022700  000402            MTPS5:  CMP     #402,R0         ;CHECK MODE 5 REGISTER
9950  017074  001401                            BEQ     TS247
 (3)  017076  104000                            EMT                     ;MODE 5 REGISTER NOT DECREMENTED BY 2
9951
9952                                    ;*****************************************************************************
 (2)                                    ;TEST 247       TEST MTPS MODE 6
 (3)                                    ;*****************************************************************************
 (2)  017100                            TS247:
9953  017100  012737  052652  000000            MOV     #52652,@#0      ;LOC. 0=52652
9954  017106  012700  000406                    MOV     #406,R0         ;R0=406
9955  017112  005037  177776                    CLR     @#PS            ;PS=0
9956  017116  106460  177372                    MTPS    -406(R0)        ;TRY MTPS W/MODE 6
9957  017122  022737  000252  177776            CMP     #252,@#PS       ;CHECK DEST. DATA
9958  017130  001401                            BEQ     MTPS6
 (2)  017132  104000                            EMT                     ;DEST. DATA INCORRECT
9959  017134  022700  000406            MTPS6:  CMP     #406,R0         ;CHECK MODE 6 REGISTER
9960  017140  001401                            BEQ     TS250
 (3)  017142  104000                            EMT                     ;MODE 6 REGISTER MODIFIED
9961
9962                                    ;*****************************************************************************
 (2)                                    ;TEST 250       TEST MTPS MODE 7
 (3)                                    ;*****************************************************************************
 (2)  017144                            TS250:
9963  017144  012737  052652  000000            MOV     #52652,@#0      ;LOC. 0=52652
9964  017152  012700  000410                    MOV     #410,R0         ;R0=410
9965  017156  005037  177776                    CLR     @#PS            ;PS=0
9966  017162  106470  177776                    MTPS    @-2(R0)         ;TRY MTPS W/MODE 7
9967  017166  022737  000105  177776            CMP     #105,@#PS       ;CHECK DEST. DATA
9968  017174  001401                            BEQ     MTPS7
 (2)  017176  104000                            EMT                     ;DESTINATION DATA INCORRECT
9969  017200  022700  000410            MTPS7:  CMP     #410,R0         ;CHECK MODE 7 REGISTER
```

```
 9970  017204  001401                        BEQ     TS251
  (3)  017206  104000                        EMT                             ;MODE 7 REGISTER MODIFIED
 9971
 9972                                ;*****************************************************************************
 9973                                ;
 9974                                ;       THESE NEXT SEVEN TESTS VERIFY THE MFPS INSTRUCTION IN ALL
 9975                                ;MODES. IN EACH TEST, A PATTERN OF ONES AND ZEROES IS MOVED TO THE
 9976                                ;PSW, AND AN MFPS INSTRUCTION MOVES THE DATA TO A LOCATION SETUP
 9977                                ;BY R0, EITHER DIRECTLY OR INDIRECTLY. CONDITIONAL BRANCHES ARE
 9978                                ;USED TO CHECK PROPER ADDRESSING AND DATA.
 9979                                ;
 9980                                ;*****************************************************************************
  (2)                               ;TEST 251       TEST MFPS INSTRUCTION
  (3)                               ;*****************************************************************************
  (2)  017210                       TS251:
 9981  017210  012737  000377  177776        MOV     #377,@#PS
 9982  017216  106700                        MFPS    R0
 9983  017220  022700  177757                CMP     #177757,R0
 9984  017224  001401                        BEQ     MFPS1
  (2)  017226  104000                        EMT                             ;MFPS FAILED
 9985
 9986  017230  005000                MFPS1:  CLR     R0
 9987  017232  012737  177777  000000        MOV     #-1,@#0
 9988  017240  005037  177776                CLR     @#PS
 9989  017244  106710                        MFPS    (R0)
 9990  017246  105737  000000                TSTB    @#0
 9991  017252  001401                        BEQ     TS252
  (3)  017254  104000                        EMT                             ;MFPS FAILED
 9992
 9993                                ;*****************************************************************************
  (2)                               ;TEST 252       TEST MFPS MODE 2
  (3)                               ;*****************************************************************************
  (2)  017256                       TS252:
 9994  017256  005000                        CLR     R0                      ;R0=0
 9995  017260  005010                        CLR     (R0)                    ;LOC. 0=0
 9996  017262  012737  000377  177776        MOV     #377,@#PS               ;SET PS=357
 9997  017270  106720                        MFPS    (R0)+                   ;TRY MFPS W/MODE 2
 9998  017272  103003                        BCC     MFPS2A                  ;BR TO ERROR IF C BIT CLEAR
 9999  017274  102402                        BVS     MFPS2A                  ;BR TO ERROR IF V BIT SET
10000  017276  001401                        BEQ     MFPS2A                  ;BR TO ERROR IF Z BIT SET
10001  017300  100401                        BMI     MFPS2B
  (1)  017302                       MFPS2A:
  (2)  017302  104000                        EMT                             ;COND. CODES INCORRECT
10002  017304  022737  000357  000000 MFPS2B: CMP     #357,@#0                ;CHECK DEST. DATA
10003  017312  001401                        BEQ     MFPS2C
  (2)  017314  104000                        EMT                             ;DEST. DATA INCORRECT
10004  017316  022700  000001        MFPS2C: CMP     #1,R0                   ;CHECK MODE 2 REGISTER
10005  017322  001401                        BEQ     TS253
  (3)  017324  104000                        EMT                             ;MODE 2 REGISTER NOT INCREMENTED 1
10006
10007                                ;*****************************************************************************
  (2)                               ;TEST 253       TEST MFPS MODE 3
  (3)                               ;*****************************************************************************
  (2)  017326                       TS253:
10008  017326  012700  000406                MOV     #406,R0                 ;R0=406
10009  017332  005037  000000                CLR     @#0                     ;LOC. 0=0
```

```
10010  017336  012737  000252  177776          MOV     #252,@#PS       ;PS=252
10011  017344  106730                          MFPS    @(R0)+          ;TRY MFPS WITH MODE 3
10012  017346  103403                          BCS     MFPS3A          ;BR TO ERROR IF C-BIT SET
10013  017350  102402                          BVS     MFPS3A          ;BR TO ERROR IF V-BIT SET
10014  017352  001401                          BEQ     MFPS3A          ;BR TO ERROR IF Z-BIT SET
10015  017354  100401                          BMI     MFPS3B
  (1)  017356                          MFPS3A:
  (2)  017356  104000                          EMT                     ;CONDITION CODES INCORRECT
10016  017360  022737  125000  000000  MFPS3B: CMP     #125000,@#0     ;CHECK DEST. DATA
10017  017366  001401                          BEQ     MFPS3C
  (2)  017370  104000                          EMT                     ;DEST DATA INCORRECT
10018  017372  020027  000410          MFPS3C: CMP     R0,#410         ;CHECK MODE 3 REGISTER.
10019  017376  001401                          BEQ     TS254
  (3)  017400  104000                          EMT                     ;MODE 3 REGISTER NOT INCREMENTED BY 2
10020
10021                                ;*************************************************************************************
  (2)                                ;TEST 254       TEST MFPS MODE 4
  (3)                                ;*************************************************************************************
  (2)  017402                        TS254:
10022  017402  012700  000002                  MOV     #2,R0           ;R0=2
10023  017406  005037  000000                  CLR     @#0             ;LOC. 0=0
10024  017412  012737  000125  177776          MOV     #125,@#PS       ;PS=125
10025  017420  106740                          MFPS    -(R0)           ;TRY MFPS W/MODE 4
10026  017422  103003                          BCC     MFPS4A          ;BR TO ERROR IF C-BIT CLEAR
10027  017424  102402                          BVS     MFPS4A          ;BR TO ERROR IF V-BIT SET
10028  017426  001401                          BEQ     MFPS4A          ;BR TO ERROR IF Z-BIT SET
10029  017430  100001                          BPL     MFPS4B
  (1)  017432                          MFPS4A:
  (2)  017432  104000                          EMT                     ;COND. CODES INCORRECT
10030  017434  022737  042400  000000  MFPS4B: CMP     #42400,@#0      ;CHECK DEST. DATA
10031  017442  001401                          BEQ     MFPS4C
  (2)  017444  104000                          EMT                     ;DEST. DATA INCORRECT
10032  017446  020027  000001          MFPS4C: CMP     R0,#1           ;CHECK MODE 4 REGISTER
10033  017452  001401                          BEQ     TS255
  (3)  017454  104000                          EMT                     ;MODE 4 REGISTER NOT DECREMENTED BY 1
10034
10035                                ;*************************************************************************************
  (2)                                ;TEST 255       TEST MFPS MODE 5
  (3)                                ;*************************************************************************************
  (2)  017456                        TS255:
10036  017456  012700  000410                  MOV     #410,R0         ;R0=410
10037  017462  012737  177777  000000          MOV     #-1,@#0         ;LOC. 0=-1
10038  017470  005037  177776                  CLR     @#PS            ;PS=0
10039  017474  106750                          MFPS    @-(R0)          ;TRY MFPS W/MODE 5
10040  017476  103403                          BCS     MFPS5A          ;BR TO ERROR IF C-BIT SET
10041  017500  102402                          BVS     MFPS5A          ;BR TO ERROR IF V-BIT SET
10042  017502  100401                          BMI     MFPS5A          ;BR TO ERROR IF N-BIT SET
10043  017504  001401                          BEQ     MFPS5B
  (1)  017506                          MFPS5A:
  (2)  017506  104000                          EMT                     ;COND. CODES INCORRECT
10044  017510  022737  000377  000000  MFPS5B: CMP     #377,@#0        ;CHECK DEST. DATA
10045  017516  001401                          BEQ     MFPS5C
  (2)  017520  104000                          EMT                     ;DEST DATA INCORRECT
10046  017522  020027  000406          MFPS5C: CMP     R0,#406         ;CHECK MODE 5 REGISTER
10047  017526  001401                          BEQ     TS256
  (3)  017530  104000                          EMT                     ;MODE 5 REGISTER NOT DECREMENTED BY 2
```

```
10048
10049                                   ;*****************************************************************************
  (2)                                   ;TEST 256       TEST MFPS MODE 6
  (3)                                   ;*****************************************************************************
  (2)  017532                           TS256:
10050  017532  012700  000401                  MOV     #401,R0         ;R0=410
10051  017536  005037  000000                  CLR     @#0             ;LOC. 0=0
10052  017542  012737  000252  177776          MOV     #252,@#PS       ;PS=252
10053  017550  106760  177377                  MFPS    -401(R0)        ;TRY MFPS W/MODE 6
10054  017554  102403                          BVS     MFPS6A          ;BR TO ERROR IF V-BIT SET
10055  017556  103402                          BCS     MFPS6A          ;BR TO ERROR IF C-BIT SET
10056  017560  001401                          BEQ     MFPS6A          ;BR TO ERROR IF Z-BIT SET
10057  017562  100401                          BMI     MFPS6B
  (1)  017564                           MFPS6A:
  (2)  017564  104000                          EMT                     ;COND. CODES INCORRECT
10058  017566  022737  000252  000000  MFPS6B: CMP     #252,@#0        ;CHECK DEST. DATA
10059  017574  001401                          BEQ     MFPS6C
  (2)  017576  104000                          EMT                     ;DEST. DATA INCORRECT
10060  017600  022700  000401           MFPS6C: CMP     #401,R0         ;CHECK DEST. REGISTER
10061  017604  001401                          BEQ     TS257
  (3)  017606  104000                          EMT                     ;DEST. DATA INCORRECT
10062
10063                                   ;*****************************************************************************
  (2)                                   ;TEST 257       TEST MFPS MODE 7
  (3)                                   ;*****************************************************************************
  (2)  017610                           TS257:
10064  017610  012700  000777                  MOV     #777,R0         ;R0=777
10065  017614  005037  000000                  CLR     @#0             ;LOC. 0=0
10066  017620  012737  000125  177776          MOV     #125,@#PS       ;PS=125
10067  017626  106770  177407                  MFPS    @-371(R0)       ;TRY MFPS W/MODE 7
10068  017632  102403                          BVS     MFPS7A          ;BR TO ERROR IF V-BIT SET
10069  017634  103002                          BCC     MFPS7A          ;BR TO ERROR IF C-BIT SET
10070  017636  001401                          BEQ     MFPS7A          ;BR TO ERROR IF Z-BIT SET
10071  017640  100001                          BPL     MFPS7B
  (1)  017642                           MFPS7A:
  (2)  017642  104000                          EMT                     ;CONDITION CODE INCORRECT
10072  017644  022737  042400  000000  MFPS7B: CMP     #42400,@#0      ;CHECK DESTINATION DATA
10073  017652  001401                          BEQ     MFPS7C
  (2)  017654  104000                          EMT                     ;DEST. DATA INCORRECT
10074  017656  022700  000777           MFPS7C: CMP     #777,R0         ;CHECK MODE 7 REGISTER
10075  017662  001401                          BEQ     TS260
  (3)  017664  104000                          EMT                     ;MODE 7 REGISTER MODIFIED
10076
10077                                   ;*****************************************************************************
10078                                   ;
10079                                   ;       THIS TEST VERIFIES THAT RESET DOES NOT CLEAR THE PSW.
10080                                   ;THE PSW IS LOADED WITH ONES, A RESET IS ISSUED, AND THE
10081                                   ;CONTENTS OF THE PSW ARE CHECKED TO VERIFY THAT THEY HAVE NOT
10082                                   ;CHANGED. THIS TEST IS EXECUTED ONLY ONCE EVERY 240 (DECIMAL)
10083                                   ;ITERATIONS OF PROGRAM.
10084                                   ;
10085                                   ;*****************************************************************************
  (2)                                   ;TEST 260       TEST THAT RESET DOES NOT CLEAR PSW
  (3)                                   ;*****************************************************************************
  (2)  017666                           TS260:
10086  017666  032737  000001  001020          BIT     #1, @#ENV       ;ARE WE RUNNING UNDER APT
```

```
  (1)  017674  001403                          BEQ     70$             ;IF NO THEN DO TEST
  (1)  017676  005737  001006                  TST     @0$PASS         ;IS THIS FIRST PASS
  (2)  017702  001011                          BNE     TS261           ;IF NO THEN SHIP TO NEXT TEST
  (1)  017704                                  70$:
10087  017704  012737  000357  177776          MOV     #357,@0PS               ;MOV ONES TO PSW
10088  017712  000005                          RESET                   ;
10089  017714  022737  000357  177776          CMP     #357,@0PS               ;PSW CORRECT?
10090  017722  001401                          BEQ     TS261
  (3)  017724  104000                          EMT                     ;RESET ALTERED PSW
10091  017726                                  REST:
10092
10093                                          ;*******************************************************************************
10094                                          ;
10095                                          ;       THE FOLLOWING TEST CHECKS THE INDEPENDENT FUNCTIONING OF BASIC
10096                                          ;DATA PATH COMPONENTS WITH USER MODE SET.
10097                                          ;
10098                                          ;*******************************************************************************
  (2)                                         ;TEST 261       TEST USER MODE R6 CAN HOLD A ONE IN EVERY POSITION
  (3)                                         ;*******************************************************************************
  (2)  017726                                  TS261:
10099  017726  052767  140000  160042          BIS     #USRM,PS                ;SET USER MODE
10100  017734  012706  000001                  MOV     #1,R6                   ;SET BIT0
10101  017740  000241                          CLC                             ;CLEAR C-BIT
10102  017742  006106                  USP1:   ROL     R6                      ;ROTATE 1 POSITION
10103  017744  103376                          BCC     USP1                    ;BR IF NOT ALL DONE
10104  017746  001404                          BEQ     USP1A                   ;BR IF NO BITS PICKED
10105  017750  042767  140000  160020          BIC     #USRM,PS                ;CLEAR USER MODE
10106  017756  104000                          EMT                     ;USER MODE R6 PICKED A BIT
10107  017760  042767  140000  160010  USP1A:  BIC     #USRM,PS                ;CLEAR USER MODE
10108
10109
10110                                          ;*******************************************************************************
10111                                          ;
10112                                          ;       THIS TEST CHECKS THE INDEPENDENT FUNCTIONING OF THE USER
10113                                          ;AND KERNEL MODE R6'S. R6 IS SETUP AND ADDRESSED IN EACH
10114                                          ;OF THE TWO MODES TO VERIFY THAT THE TWO R6'S ARE INDEPENDENT
10115                                          ;OF EACH OTHER.
10116                                          ;
10117                                          ;*******************************************************************************
  (2)                                         ;TEST 262       TEST INDEPENDENCE OF USER AND KERNEL MODE R6'S
  (3)                                         ;*******************************************************************************
  (2)  017766                                  TS262:
10118  017766  052767  140000  160002          BIS     #USRM,PS                ;SET USER MODE
10119  017774  012706  177777                  MOV     #-1,R6                  ;SET USER R6 TO ALL ONES
10120  020000  022706  177777                  CMP     #-1,R6                  ;READ AND CHECK USER R6
10121  020004  001404                          BEQ     USP2                    ;BR IF NO ERROR
10122  020006  042767  140000  157762          BIC     #USRM,PS                ;CLEAR USER MODE
10123  020014  104000                          EMT                     ;USER R6 WILL NOT HOLD ALL ONES
10124  020016  042767  140000  157752  USP2:   BIC     #USRM,PS                ;SET KERNEL MODE
10125  020024  022706  177777                  CMP     #-1,R6                  ;KERNEL MODE R6 ADDR. FROM USER MODE?>>
10126  020030  001001                          BNE     USP3
  (2)  020032  104000                          EMT                     ;DUAL ADDRESSING ERROR USER/KERNEL R6
10127  020034  005006                  USP3:   CLR     R6                      ;CLEAR KERNEL MODE SP
10128  020036  052767  140000  157732          BIS     #USRM,PS                ;SET USER MODE
10129  020044  022706  177777                  CMP     #-1,R6                  ;CHECK USER R6 NOT ADDR. FROM KERNEL MODE
10130  020050  042767  140000  157720          BIC     #USRM,PS                ;CLEAR USER MODE
```

```
10131  020056  001401                        BEQ     USP4                    ;BR IF NO ERROR
10132  020060  104000                        EMT                     ;DUAL ADDRESSING ERROR OR SEQUENCE ERROR
10133  020062  012706  001000        USP4:   MOV     #STBOT,R6               ;RESTORE SP USER
10134  020066  042767  140000  157702        BIC     #USRM,PS                ;SET KERNEL MODE
10135  020074  012706  001000                MOV     #STBOT,R6               ;RESTORE SP KERNEL
10136
10137                                  ;*************************************************************************
10138                                  ;
10139                                  ;       THESE NEXT TWO TESTS VERIFY MFPI AND MTPI INSTRUCTIONS
10140                                  ;WITH R6 IN MODE 0.
10141                                  ;
10142                                  ;*************************************************************************
  (2)                                 ;TEST 263       TEST MFPI WITH R6 IN MODE 0
  (3)                                 ;*************************************************************************
  (2)  020100                         TS263:
10143  020100  012706  001000                MOV     #STBOT,R6               ;INITIALIZE KERNEL STACK POINTER
10144  020104  012767  140000  157664        MOV     #USRM,PS                ;SET USER MODE.PREVIOUS KERNEL
10145  020112  012706  000600                MOV     #USESTK,R6              ;INITIALIZE USER STACK POINTER
10146  020116  006506                        MFPI    R6                      ;TRY MFPI WITH MODE 0
10147  020120  022767  140000  157650        CMP     #140000,PS              ;CHECK PSW
10148  020126  001404                        BEQ     MFPIO                   ;BR IF NO ERROR
10149  020130  042767  140000  157640        BIC     #USRM,PS                ;CLEAR USER MODE
10150  020136  104000                        EMT                     ;INCORRECT PSW FROM MFPI
10151  020140  042767  140000  157630  MFPIO: BIC     #USRM,PS                ;CLEAR USER MODE
10152  020146  022767  001000  160422        CMP     #STBOT,USESTK-2         ;CHECK DATA ON STACK
10153  020154  001401                        BEQ     MFPIOA                  ;BR IF NO ERROR
10154  020156  104000                        EMT                     ;INCORRECT DATA FROM MFPI
10155  020160                         MFPIOA:
10156
10157                                  ;*************************************************************************
  (2)                                 ;TEST 264       TEST MTPI WITH R6 IN MODE 0
  (3)                                 ;*************************************************************************
  (2)  020160                         TS264:
10158  020160  005067  157612                CLR     PS                      ;SET KERNEL MODE
10159  020164  005006                        CLR     R6                      ;INITIALIZE KERNEL R6
10160  020166  012767  140000  157602        MOV     #USRM,PS                ;SET USER MODE/PREVIOUS KERNEL
10161  020174  012706  000600                MOV     #USESTK,R6              ;INITIALIZE USER STACK POINTER
10162  020200  012746  001000                MOV     #STBOT,-(R6)            ;SET UP TARGET DATA
10163  020204  006606                        MTPI    R6                      ;TRY MODE 0 MTPI
10164  020206  022767  140000  157562        CMP     #USRM,PS                ;CHECK PSW
10165  020214  001404                        BEQ     MTPIO                   ;BR IF NO ERROR
10166  020216  042767  140000  157552        BIC     #USRM,PS                ;CLEAR USER MODE
10167  020224  104000                        EMT                     ;PS INCORRECT FOLLOWING MTPI
10168  020226  005067  157544        MTPIO:  CLR     PS                      ;SET KERNEL MODE
10169  020232  020627  001000                CMP     R6,#STBOT               ;CHECK TARGET DATA
10170  020236  001401                        BEQ     TS265
  (3)  020240  104000                        EMT                     ;DATA INCORRECT FOLLOWING MTPI
10171
10172
10173
10174                                  ;*************************************************************************
10175                                  ;
10176                                  ;THE FOLLOWING TEST VERIFIES THAT NO DUAL ADDRESSING OF THE GENERAL
10177                                  ;REGISTERS OCCURS. ALL REGISTERS ARE CLEARED, AND A UNIQUE BIT IS SET
10178                                  ;IN EACH. CMP INSTRUCTIONS CHECK THAT ONLY ONE BIT IS SET IN EACH
10179                                  ;REGISTER.
```

```
10180                                   ;
10181                                   ;*****************************************************************************
  (2)                                   ;TEST 265       DUAL REGISTER ADDRESSING TEST
  (3)                                   ;*****************************************************************************
  (2)  020242                           TS265:
10182  020242  005000                   BITCLR: CLR     R0              ;INITIALIZE ALL REGISTERS
10183  020244  005001                           CLR     R1
10184  020246  005002                           CLR     R2
10185  020250  005003                           CLR     R3
10186  020252  005004                           CLR     R4
10187  020254  005005                           CLR     R5
10188  020256  005006                           CLR     R6
10189  020260  052700  000001         BITSET: BIS     #1,R0           ;SET R0=1
10190  020264  052701  000002                 BIS     #2,R1           ;R1=2
10191  020270  052702  000004                 BIS     #4,R2           ;R2=4
10192  020274  052703  000010                 BIS     #10,R3          ;R3=10
10193  020300  052704  000020                 BIS     #20,R4          ;R4=20
10194  020304  052705  000040                 BIS     #40,R5          ;R5=40
10195  020310  052706  000100                 BIS     #100,R6         ;R6=100
10196  020314  022706  000100         BITCHK: CMP     #100,R6         ;TEST THAT NO DUAL ADDRESSING OCCURRED
10197  020320  001022                         BNE     DAERR           ;BR TO ERROR HALT IF ANY OTHER BITS ARE SET
10198  020322  022705  000040                 CMP     #40,R5
10199  020326  001017                         BNE     DAERR
10200  020330  022704  000020                 CMP     #20,R4
10201  020334  001014                         BNE     DAERR
10202  020336  022703  000010                 CMP     #10,R3
10203  020342  001011                         BNE     DAERR
10204  020344  022702  000004                 CMP     #4,R2
10205  020350  001006                         BNE     DAERR
10206  020352  022701  000002                 CMP     #2,R1
10207  020356  001003                         BNE     DAERR
10208  020360  022700  000001                 CMP     #1,R0
10209  020364  001401                         BEQ     BITCON
  (1)  020366                           DAERR:
  (2)  020366  104000                         EMT                     ;DUAL ADDRESSING ERROR
10210  020370  012702  001004         BITCON: MOV     #$TESTN,R2      ;RESTORE POINTER
10211
10212                                   ;*****************************************************************************
10213                                   ;       THIS TEST VERIFIES THAT THE UPPER BYTE OF THE PSW IS NOT AFFECTED
10214                                   ;WHEN THE PRIORITY LEVEL OR CC'S ARE CHANGED. ALL BITS ARE
10215                                   ;INITIALLY SET IN THE PSW, AND THE LOW BYTE IS CLEARED. A BIT
10216                                   ;INSTRUCTION VERIFIES THE DATA.
10217                                   ;
10218                                   ;*****************************************************************************
  (2)                                   ;TEST 266       TEST BYTE INSTRUCTION ON PSW
  (3)                                   ;*****************************************************************************
  (2)  020374                           TS266:
10219  020374  052737  170357  177776         BIS     #170357,@#PS    ;SET ALL POSSIBLE BITS IN PSW
10220  020402  105037  177776                 CLRB    @#PS            ;CLR PR LEVEL AND CC'S
10221  020406  013700  177776                 MOV     @#PS,R0         ;COPY CONTENTS OF PSW
10222  020412  032700  170000                 BIT     #170000,R0      ;TEST THAT UPPER BYTE IS UNAFFECTED
10223  020416  001003                         BNE     BTCON           ;CONTINUE IF OK
10224  020420  005037  177776         BTERR:  CLR     @#PS            ;RETURN TO KERNEL MODE
10225  020424  104000                         EMT                     ;BYTE INSTRUCTION ALTERED PSW
10226  020426  005037  177776         BTCON:  CLR     @#PS            ;RETURN TO KERNEL MODE
10227
```

```
10228                                   ;*****************************************************************************
10229                                   ;
10230                                   ;       THIS TEST VERIFIES THAT A JMP INSTRUCTION DOES NOT ALTER THE
10231                                   ;CONDITION CODES IN THE PSW. THE CC'S ARE PRESET,THE JMP IS
10232                                   ;EXECUTED, AND CONDITIONAL BRANCHES VERIFY THE STATE OF THE CC'S.
10233                                   ;
10234                                   ;*****************************************************************************
  (2)                                   ;TEST 267       TEST THAT JMP INSTRUCTION DOES NOT AFFECT CONDITION CODES
  (3)                                   ;*****************************************************************************
  (2)  020432                           TS267:
10235  020432  000277                          SCC
10236  020434  000252                          +CLN!CLV                ;CC=0101
10237  020436  000167  00000C                  JMP     JMPT            ;JUMP TO TEST PSW
10238  020442  100403                  JMPT:   BMI     JMPERR          ;BR TO ERROR HALT IF N-BIT IS SET
10239  020444  001002                          BNE     JMPERR          ;BR TO ERROR HALT IF Z-BIT IS CLEAR
10240  020446  102401                          BVS     JMPERR          ;BR TO ERROR HALT IF V-BIT IF SET
10241  020450  103401                          BCS     TS270
  (2)  020452                           JMPERR:
  (3)  020452  104000                          EMT                     ;JMP INSTRUCTION AFFECTED CC'S
10242                                   ;*****************************************************************************
10243                                   ;
10244                                   ;       THIS TEST VERIFIES THE SET AND CLEAR CONDITION CODE INSTRUCTIONS.
10245                                   ;THE TEST CONSISTS OF TWO ROUTINES, ONE TO TEST ALL CLEAR CC
10246                                   ;INSTRUCTIONS, AND THE SECOND TO TEST ALL SET CC INSTRUCTIONS. ALL
10247                                   ;POSSIBLE COMBINATIONS OF CONDITION CODES ARE TESTED, INCLUDING NOP'S.
10248                                   ;       TO TEST THE CLEAR CC INSTRUCTIONS, ALL CONDITION CODES ARE
10249                                   ;INITIALLY SET. THE INSTRUCTION IS EXECUTED, AND THE PSW IS CHECKED
10250                                   ;TO VERIFY THE PROPER COMBINATION OF CONDITION CODES.
10251                                   ;       TO TEST THE SET CC INSTRUCTIONS, THE CONDITION CODES ARE
10252                                   ;INITIALLY CLEARED, AND ONLY THE REQUIRED BITS ARE SET BY THE SET CC
10253                                   ;INSTRUCTION. THE CONTENTS OF THE PSW ARE CHECKED TO VERIFY THAT
10254                                   ;ONLY THE REQUIRED BITS WERE SET.
10255                                   ;
10256                                   ;*****************************************************************************
  (2)                                   ;TEST 270       TEST SET CC AND CLEAR CC INSTRUCTIONS
  (3)                                   ;*****************************************************************************
  (2)  020454                           TS270:
10257  020454  012767  000240  000024          MOV     #240,CC3        ;INITIALIZE CLR CC INSTRUCTION CODES
10258  020462  012767  000017  000032          MOV     #17,CC2         ;INITIALIZE OCTAL MAP
10259  020470  012767  000261  000074          MOV     #261,SC3        ;INITIALIZE SET CC INSTRUCTION CODES
10260  020476  012767  000001  000102          MOV     #1,SC4          ;INITIALIZE OCTAL MAP
10261  020504  000277                  CLRCD:  SCC                     ;SET ALL CONDITION CODES
10262  020506  000000                  CC3:    0                       ;CONDITION CODE INSTRUCTION
10263  020510  013704  177776                  MOV     @#PS,R4         ;COPY THE PSW
10264  020514  042704  177760                  BIC     #177760,R4      ;ISOLATE CONDITION CODES
10265  020520  022704                          CMP     (PC)+,R4        ;CHECK THAT PROPER CC'S WERE CLEARED
10266  020522  000000                  CC2:    0                       ;OCTAL REPRESENTATION OF CC'S
10267  020524  001401                          BEQ     CON1
  (2)  020526  104000                          EMT                     ;CLEAR CC INSTRUCTION FAILED
10268  020530  005367  177766          CON1:   DEC     CC2             ;SET NEXT OCTAL MAP OF CC'S
10269  020534  005267  177746                  INC     CC3             ;GET NEXT CLEAR CC INSTRUCTION
10270  020540  026727  177742  000257          CMP     CC3,#257        ;TEST FOR CCC INSTRUCTION
10271  020546  003756                          BLE     CLRCD           ;GO TEST NEXT INSTRUCTION IF NOT FOUND
10272  020550  026727  177732  000260          CMP     CC3,#260        ;CHECK FOR NOP=260
10273  020556  001004                          BNE     SETCD           ;GO TEST SET CC INSTRUCTIONS
10274  020560  012767  000017  177734          MOV     #17,CC2         ;SET OCTAL MAP TO TEST NOP
```

```
10275  020566  000746                        BR      CLRCD           ;GO TEST NOP
10276  020570  000257                SETCD:  CCC                     ;CLEAR ALL CONDITION CODES
10277  020572  000000                SC3:    0                       ;CONDITION CODE INSTRUCTION
10278  020574  013704  177776                MOV     @#PS,R4         ;COY PSW
10279  020600  042704  177760                BIC     #177760,R4      ;CLEAR AWAY UNWANTED BITS
10280  020604  022704                        CMP     (PC)+,R4        ;CHECK THAT PROPER CC'S WERE SET
10281  020606  000000                SC4:    0                       ;OCTAL REPRESENTATION OF CC'S
10282  020610  001401                        BEQ     CON2
  (1)  020612                        CCERR:
  (2)  020612  104000                        EMT                     ;SET CC FAILED OR SEQUENCE ERROR
10283  020614  005267  177766        CON2:   INC     SC4             ;SET NEXT OCTAL MAP
10284  020620  005267  177746                INC     SC3             ;PREPARE NEXT SET CC INSTRUCTION
10285  020624  026727  177742  000277        CMP     SC3,#277        ;FINISHED?
10286  020632  003756                        BLE     SETCD           ;BR IF NO
10287  020634  000167  000006                JMP     MORO            ;JUMP TO NEXT TESTS
```

H9

CJKL5B0 LCP-5 CPU CLSTR DIAG  MACY11 30(1046)  07-JAN-85  09:28  PAGE 10-12
CJKL5B.P11  07-JAN-85 09:05  T270  TEST SET CC AND CLEAR CC INSTRUCTIONS  SEQ 0111

```
                                        ;*******************************************************************************
                                        .SBTTL  TEST INSTRUCTIONS USING SAME REGISTER FOR SOURCE & DESTINATION
                                        ;
                                        ;IN AUTO INCREMENT (DECREMENT) MODES AND
                                        ;AUTO INCREMENT (DECREMENT) DEFERRED MODES,
                                        ;CONTENTS OF THE REGISTER IN USED ARE
                                        ;INCREMENTED (DECREMENTED) BY 2
                                        ;BEFORE USED AS THE SOURCE OPERAND.
                                        ;
020640  000000  000000  000000  A:      .WORD 0,0,0
020646                                  MOR0:
                                        ;*******************************************************************************
                                        ;TEST 271       TEST AUTO-INCREMENT MODE, USING R0
                                        ;*******************************************************************************
020646                                  TS271:
020646  005037  020640                          CLR     @#A             ;CLEAR LOC A
020652  012700  020640                          MOV     #A,R0           ;R0 STORES ADDR OF A
020656  060020                                  ADD     R0,(R0)+        ;CHECK THAT R0 IS INCR BY 2 BEFORE
                                                                        ;BEING USED AS THE SOURCE OPERAND
020660  022700  020642                          CMP     #A+2,R0         ;R0 INCR BY 2?
020664  001401                                  BEQ     MOR1
020666  104000                                  EMT                     ;R0 WAS NOT INCREMENTED BY 2
                                                                        ;
020670  022737  020642  020640  MOR1:           CMP     #A+2,@#A        ;CHECK CONTENT OF R0 WAS INCR BY 2 BEFORE
                                                                        ;BEING USED IN THE "ADD" INSTR
                                                                        ;LOC A CONTAINS (A+2)?
020676  001401                                  BEQ     TS272
020700  104000                                  EMT                     ;WRONG SUM IN LOC A

                                        ;*******************************************************************************
                                        ;TEST 272       AUTO-DECREMENT MODE, USING R0
                                        ;*******************************************************************************
020702                                  TS272:
020702  005037  020640                          CLR     @#A             ;CLEAR LOC A
020706  012700  020642                          MOV     #A+2,R0         ;R0 STORES ADDR OF A+2
020712  060040                                  ADD     R0,-(R0)        ;CHECK THAT R0 IS DECR BY 2 BEFORE
                                                                        ;BEING USED AS THE SOURCE OPERAND
020714  022700  020640                          CMP     #A,R0           ;R0 DECR BY 2?
020720  001401                                  BEQ     MOR2
020722  104000                                  EMT                     ;R0 WAS NOT DECREMENTED BY 2
                                                                        ;
020724  022737  020640  020640  MOR2:           CMP     #A,@#A          ;CONTENT OF R0 WAS DECR BY 2 BEFORE
                                                                        ;BEING USED IN THE "ADD" INSTR
                                                                        ;LOC A CONTAINS (R0)
020732  001401                                  BEQ     TS273
020734  104000                                  EMT                     ;WRONG SUM IN LOC A
                                                                        ;
                                        ;*******************************************************************************
                                        ;TEST 273       TEST AUTO-INCREMENT DEFERRED MODE, USING R0
                                        ;*******************************************************************************
020736                                  TS273:
020736  005037  020640                          CLR     @#A             ;CLEAR LOC A
020742  005037  020644                          CLR     @#A+4           ;CLEAR LOC A+4
020746  012737  020640  020642                  MOV     #A,@#A+2        ;STORE ADDR A IN LOC A+2
020754  012700  020642                          MOV     #A+2,R0         ;R0 STORES ADDR A+2
020760  060030                                  ADD     R0,@(R0)+       ;CHECK THAT R0 IS INCR BY 2 BEFORE
```

```
                                                                        ;BEING USED AS THE SOURCE OPERAND
020762  022700  020644                  CMP     #A+4,R0                 ;R0 INCR BY 2?
020766  001401                          BEQ     MOR3
020770  104000                          EMT                             ;R0 WAS NOT INCREMENTED BY 2
                                                                        ;
020772  022737  020640  020642  MOR3:   CMP     #A,@#A+2                ;LOC A+2 STILL STORES ADDR A?
021000  001401                          BEQ     MOR4
021002  104000                          EMT                             ;LOC A+2 STORES WRONG DATA
                                                                        ;
021004  022737  020644  020640  MOR4:   CMP     #A+4,@#A                ;CHECK CONTENT OF R0 WAS INCR BY 2 BEFORE
                                                                        ;BEING USED IN THE "ADD" INSTR
021012  001401                          BEQ     MOR5
021014  104000                          EMT                             ;LOC A STORES WRONG DATA
                                                                        ;
021016  005737  020644          MOR5:   TST     @#A+4                   ;LOC A+4 STILL STORES 0?
021022  001401                          BEQ     TS274
021024  104000                          EMT                             ;LOC A+4 DID NOT STAY CLEAR

                                ;*******************************************************************************
                                ;TEST 274       TEST AUTO-DECREMENT DEFERRED, USING R0
                                ;*******************************************************************************
021026                          TS274:
021026  005037  020640                  CLR     @#A                     ;CLEAR LOC A
021032  005037  020644                  CLR     @#A+4                   ;CLEAR LOC A+4
021036  012700  020644                  MOV     #A+4,R0                 ;R0 STORES ADDR A+4
021042  012737  020640  020642          MOV     #A,@#A+2                ;STORE ADDR A IN LOC A+2
021050  060050                          ADD     R0,@-(R0)               ;CHECK THAT R0 IS DECR BY 2 BEFORE
                                                                        ;BEING USED AS THE SOURCE OPERAND
021052  022700  020642                  CMP     #A+2,R0                 ;R0 DECREMENTED BY 2?
021056  001401                          BEQ     MOR6
021060  104000                          EMT                             ;R0 WAS NOT DECREMENTED BY 2
                                                                        ;
021062  022737  020642  020640  MOR6:   CMP     #A+2,@#A                ;CHECK CONTENT OF R0 WAS DECR BY 2 BEFORE
                                                                        ;BEING USED IN THE "ADD" INSTR
021070  001401                          BEQ     MOR7
021072  104000                          EMT                             ;LOC A STORES WRONG DATA


                                                                        ;
021074  022737  020640  020642  MOR7:   CMP     #A,@#A+2                ;LOC A+2 STILL STORES A?
021102  001401                          BEQ     MOR8
021104  104000                          EMT                             ;LOC A+2 STORES WRONG DATA
                                                                        ;
021106  005737  020644          MOR8:   TST     @#A+4                   ;LOC A+4 STILL STORES 0?
021112  001401                          BEQ     TS275
021114  104000                          EMT                             ;LOC A+4 DID NOT STAY CLEAR
                                                                        ;
```

```
                                        ;*******************************************************************************
                                        .SBTTL  INSTRUCTION USING PC AS SOURCE REGISTER
                                        ;
                                        ;IN INDEX, INDEX DEFERRED, RELATIVE, AND
                                        ;RELATIVE DEFERRED MODES, DESTINATION WILL CONTAIN
                                        ;THE PC COUNT OF THE CURRENT INSTRUCTION +4.
                                        ;
                                        ;*******************************************************************************
                                        ;TEST 275       TEST PC AS SOURCE IN MODE 0, USING R0
                                        ;*******************************************************************************
  (2) 021116                            TS275:
  (3) 021116  012700  177777                    MOV     #-1,R0          ;SET ALL 1 IN R0
  (2) 021122  010700                    PCN01:  MOV     PC,R0           ;STORES PC IN R0
  (3) 021124  022700  021124                    CMP     #PCN01+2,R0     ;R0 STORES PC+2?
  (3) 021130  001401                            BEQ     TS276
  (3) 021132  104000                            EMT                     ;R0 STORED WRONG VALUE

                                        ;*******************************************************************************
                                        ;TEST 276       TEST PC AS SOURCE IN MODE 6, USING R0
                                        ;*******************************************************************************
  (2) 021134                            TS276:
  (3) 021134  012700  020640                    MOV #A,R0               ;R0 STORES ADDR A
  (3) 021140  010760  000004            PCN2:   MOV PC,4(R0)            ;EFFECTIVE ADDR IS A+4
  (3) 021144  022737  021144  020644            CMP     #PCN2+4,@#A+4   ;LOC A+4 STORES PC+4?
  (3) 021152  001401                            BEQ     TS277
  (3) 021154  104000                            EMT                     ;LOC A+4 STORED WRONG VALUE

                                        ;*******************************************************************************
                                        ;TEST 277       TEST PC AS SOURCE IN MODE 7, USING R0
                                        ;*******************************************************************************
  (2) 021156                            TS277:
  (3) 021156  012737  020640  020644            MOV     #A,@#A+4        ;LOC A+4 STORES ADDR A
  (3) 021164  012700  020640                    MOV     #A,R0           ;R0 STORES ADDR A
  (3) 021170  010770  000004            PCN3:   MOV     PC,@4(R0)       ;EFFECTIVE ADDR IS A
  (3) 021174  022737  021174  020640            CMP     #PCN3+4,@#A     ;LOC A STORES PC+4?
  (3) 021202  001401                            BEQ     TS300
  (3) 021204  104000                            EMT                     ;LOC A STORED WRONG VALUE

                                        ;*******************************************************************************
                                        ;TEST 300       TEST PC AS SOURCE IN RELATIVE DEFERRED MODE ,USING R0
                                        ;*******************************************************************************
  (2) 021206                            TS300:
  (3) 021206  012737  020642  020640            MOV     #A+2,@#A        ;LOC A STORES ADDR A+2
  (3) 021214  010777  177420            PCN4:   MOV     PC,@A           ;EFFECTIVE ADDR IS A+2
  (3) 021220  022737  021220  020642            CMP     #PCN4+4,@#A+2   ;LOC A+2 STORES PC+4?
  (3) 021226  001401                            BEQ     TS301
  (3) 021230  104000                            EMT                     ;LOC A+2 STORED WRONG VALUE

                                        ;*******************************************************************************
                                        ;TEST 301       TEST PC AS SOURCE IN RELATIVE MODE ,USING R0
                                        ;*******************************************************************************
  (2) 021232                            TS301:
  (3) 021232  005037  020640                    CLR     @#A             ;CLEAR A
  (3) 021236  010767  177376            PCN5:   MOV     PC,A            ;EFFECTIVE ADDR IS A
  (3) 021242  022737  021242  020640            CMP     #PCN5+4,@#A     ;LOC A STORES PC+4?
  (3) 021250  001401                            BEQ     TS302
```

K9



```
  (3)  021252  104000                          EMT                     ;LOCATION A STORED WRONG VALUE
                                        ;*******************************************************************************
                                        ;THIS TESTS THE MOVE FROM PROCESSOR TYPE INSTRUCTION(MFPT)
                                        ;UPON EXECUTION R0 WILL RECIEVE THE PROCESSOR MODEL CODE
                                        ;WHICH IS "000003" FOR THE DCF11-AA
                                        ;*******************************************************************************
                                        ;*******************************************************************************
  (2)                                   ;TEST 302        TEST MFPT
  (3)                                   ;*******************************************************************************
  (2)  021254                           TS302:
                000007                  MFPT=000007
       021254  012706  001000                  MOV     #STBOT,SP       ;INITIALIZE STACK POINT IN CASE OF TRAP
       021260  000007                          MFPT                    ;GET MODEL CODE.IF THIS TRAPS AN ERROR WILL BE REPORTED
       021262  022700  000003                  CMP     #3,R0           ;CHECK IF CORRECT CODE RETURNED
       021266  001401                          BEQ     TS303
  (3)  021270  104000                          EMT                     ;WRONG CODE RETURNED

                                        ;*******************************************************************************
                                        .SBTTL  THE NEXT THREE TESTS EXERCISE MASKING ACTION OF MICROCODES.
                                        ;*******************************************************************************
  (2)                                   ;TEST 303        TEST SUB INSTRUCTION, SM=0, DM=2
  (3)                                   ;*******************************************************************************
  (2)  021272                           TS303:
       021272  012737  052525  000000          MOV     #052525,@#0     ;SET UP LOC 0
       021300  012701  050505                  MOV     #050505,R1      ;SET UP R1
       021304  005000                          CLR     R0              ;CLEAR R0
       021306  160120                          SUB     R1,(R0)+        ;SUBTRACTION, SM=0,DM=2
       021310  022737  002020  000000          CMP     #2020,@#0       ;CHECK DIFFERENCE AT LOC 0
       021316  001401                          BEQ     TS304
  (3)  021320  104000                          EMT                     ;WRONG RESULT FROM SUBTRACTION

                                        ;*******************************************************************************
  (2)                                   ;TEST 304        TEST MFPD WITH R0, IN MODE 2
  (3)                                   ;*******************************************************************************
  (2)  021322                           TS304:
       021322  012737  052525  000000          MOV     #052525,@#0     ;SET UP LOC 0
       021330  005000                          CLR     R0              ;CLEAR R0
       021332  012767  170000  156436          MOV     #170000,PS      ;SET USER MODE ON, CURRENT & PREVIOUS
       021340  012706  000600                  MOV     #USESTK,R6      ;SET USER STACK POINTER
       021344  106520                          MFPD    (R0)+           ;MODE 2, MFPD
       021346  005067  156424                  CLR     PS              ;SET KERNEL MODE
       021352  022767  052525  157216          CMP     #052525,USESTK-2        ;CHECK DATA ON STACK
       021360  001401                          BEQ     TS305
  (3)  021362  104000                          EMT                     ;INCORRECT DATA FROM MFPD

                                        ;*******************************************************************************
  (2)                                   ;TEST 305        TEST MTPD WITH R0, IN MODE 2
  (3)                                   ;*******************************************************************************
  (2)  021364                           TS305:
       021364  012767  170000  156404          MOV     #170000,PS      ;SET USER MODE ON, CURRENT & PREVIOUS
       021372  012706  000600                  MOV     #USESTK,R6      ;SET USER STACK POINTER
       021376  012746  125252                  MOV     #125252,-(R6)   ;PUSH DATA IN USER STACK
       021402  012737  000000  000000          MOV     #0,@#0          ;CLEAR LOC 0
       021410  005000                          CLR     R0              ;CLEAR R0
```

```
10444  021412  106620                    MTPD    (R0)+           ;MODE 2, MTPD
10445  021414  005067  156356            CLR     PS              ;SET KERNEL MODE
10446  021420  022737  125252  000000    CMP     #125252,@#0     ;CHECK DATA ON LOC 0
10447  021426  001464                    BEQ     TESTN1
  (2)  021430  104000                    EMT                     ;INCORRECT DATA FROM MTPD
10448
10449  021432  000402            BRTAB:  BR      .+6
10450  021434  001002                    BNE     .+6
10451  021436  001402                    BEQ     .+6
10452  021440  002002                    BGE     .+6
10453  021442  002402                    BLT     .+6
10454  021444  003002                    BGT     .+6
10455  021446  003402                    BLE     .+6
10456  021450  100002                    BPL     .+6
10457  021452  100402                    BMI     .+6
10458  021454  101002                    BHI     .+6
10459  021456  101402                    BLOS    .+6
10460  021460  102002                    BVC     .+6
10461  021462  102402                    BVS     .+6
10462  021464  103002                    BCC     .+6             ;SAME AS BHIS
10463  021466  103402                    BCS     .+6             ;SAME AS BLO
10464
10465          000002            .RADIX  2
10466  021470  177777            YNTAB:  1111111111111111        ;BR
10467  021472  170360                    1111000011110000        ;BNE:  Z=0
10468  021474  007417                    0000111100001111        ;BEQ:  Z=1
10469  021476  146063                    1100110000110011        ;BGE:  N XOR V =0
10470  021500  031714                    0011001111001100        ;BLT:  N XOR V =1
10471  021502  140060                    1100000000110000        ;BGT:  Z+(N XOR V) =0
10472  021504  037717                    0011111111001111        ;BLE:  Z+(N XOR V) =1
10473
10474  021506  177400                    1111111100000000        ;BPL:  N=0
10475  021510  000377                    0000000011111111        ;BMI:  N=1
10476  021512  120240                    1010000010100000        ;BHI:  C+Z=0
10477  021514  057537                    0101111101011111        ;BLOS: C+Z=1
10478  021516  146314                    1100110011001100        ;BVC:  V=0
10479  021520  031463                    0011001100110011        ;BVS:  V=1
10480  021522  125252                    1010101010101010        ;BCC:  C=0
10481  021524  052525                    0101010101010101        ;BCS:  C=1
10482          000010            .RADIX  8
10483
10484                            ;*****************************************************************************************
10485                            ; THE FOLLOWING ARE SPECIAL CPU TRAP
10486                            ;HANDLERS TO TRAP AND REPORT SPECIAL TRAPS.
10487                            ;
10488                            ;*****************************************************************************************
10489
10490  021526                    TO4:
  (2)  021526  104000                    EMT                     ;TRAPPED THRU LOC. 4
10491  021530                    TO10:
  (2)  021530  104000                    EMT                     ;TRAPPED THRU LOC. 10
10492  021532                    TO14:
  (2)  021532  104000                    EMT                     ;TRAPPED THRU LOC. 14
10493  021534                    TO20:
  (2)  021534  104000                    EMT                     ;TRAPPED THRU LOC. 20
10494  021536                    TO30:
```

```
  (2)  021536  104000                        EMT                         ;TRAPPED THRU LOC. 30
10495  021540                        T034:
  (2)  021540  104000                        EMT                         ;TRAPPED THRU LOC. 34
10496  021542                        T040:
  (2)  021542  104000                        EMT                         ;TRAPPED THRU LOC. 40
10497  021544                        T0114:
  (2)  021544  104000                        EMT                         ;TRAPPED THRU LOC. 114
10498  021546                        T0244:
  (2)  021546  104000                        EMT                         ;TRAPPED THRU LOC. 244
10499  021550                        T0250:
  (2)  021550  104000                        EMT                         ;TRAPPED THRU LOC. 250
10503                                .SBTTL ** STARTING OF TRAP TEST **
10504
10505                                ;SPECIAL CASE OF ODD;.EVEN .BYTE AND REGISTER 6
10506       000000                   HERE=0
10507
10508  021552  000000                K1:     0
10509  021554  000000                K2:     0
10510  021556  000000                K3:     0
10511  021560  000000                K4:     0
10512  021562  000000                K5:     0
10513  021564  000000                K6:     0
10514  021566  052525                K7:     052525
10515  021570  052400                K10:    052400
10516  021572  000000                K11:    0
10517  021574  000000                K12:    0
10518  021576  000176                SWR:    176
10519
10520  021600  032737  000001  001020  TESTN1: BIT     #1,@#$ENV
10521  021606  001403                        BEQ     1$
10522  021610  012767  001022  177760        MOV     #$SWREG,SWR
10523  021616                        1$:
  (3)                                ;*****************************************************************************
  (2)                                ;TEST 306       TEST AUTO INCREMENT AND DECREMENT OF R6 FOR WORD AND BYTES
  (3)                                ;*****************************************************************************
  (2)  021616                        TS306:
10524  021616  005006                        CLR     %6
10525  021620  112667  156154              MOVB    (6)+,HERE       ;SIX SHOULD INCREMENT BY TWO
10526  021624  020627  000002              CMP     %6,#2
10527  021630  001401                        BEQ     BR1
  (2)  021632  104000                        EMT                     ;R6 DID NOT AUTO INCREMENT BY TWO
10528
10529  021634  012706  001000        BR1:    MOV     #1000,%6
10530  021640  114627  000000              MOVB    -(6),@HERE      ;SHOULD DECREMENT BY TWO
10531  021644  020627  000776              CMP     %6,#776
10532  021650  001401                        BEQ     BR2
  (2)  021652  104000                        EMT                     ;R6 DID NOT AUTO DECREMENT BY 2
10533
10534  021654  005006        BR2:    CLR     %6
10535  021656  112626                MOVB    (6)+,(6)+       ;DOUBLES AUTO INCREMENT OF R6
10536  021660  020627  000004              CMP     %6,#4
10537  021664  001401                        BEQ     BR3
  (2)  021666  104000                        EMT                     ;WRONG AUTO INCREMENT OF R6
10538
10539  021670  005006        BR3:    CLR     %6
10540  021672  005004                CLR     %4
```

```
10541  021674  122624                      CMPB    (6)+,(4)+       ;TEST INCREMENT OF R6
10542  021676  020627  000002              CMP     %6,#2
10543  021702  001401                      BEQ     BR4
  (2)  021704  104000                      EMT                     ;WRONG INCREMENT OF R6
10544
10545  021706  005006              BR4:    CLR     %6
10546  021710  005004                      CLR     %4
10547  021712  122426                      CMPB    (4)+,(6)+       ;TEST INCREMENT OF R6
10548  021714  020627  000002              CMP     %6,#2
10549  021720  001401                      BEQ     BR5
  (2)  021722  104000                      EMT                     ;WRONG INCREMENT OF R6
10550
10551  021724  005006              BR5:    CLR     %6
10552  021726  005004                      CLR     %4
10553  021730  122624                      CMPB    (6)+,(4)+       ;TEST INCREMENT OF R4
10554  021732  020427  000001              CMP     %4,#1
10555  021736  001401                      BEQ     BR6
  (2)  021740  104000                      EMT                     ;WRONG INCREMENT OF R4
10556  021742  005006              BR6:    CLR     %6
10557  021744  005004                      CLR     %4
10558  021746  122426                      CMPB    (4)+,(6)+       ;TEST INCREMENT OF R6
10559  021750  020627  000002              CMP     %6,#2
10560  021754  001401                      BEQ     BR7
  (2)  021756  104000                      EMT                     ;WRONG INCREMENT OF R6
10561
10562  021760  005006              BR7:    CLR     %6
10563  021762  005004                      CLR     %4
10564  021764  122426                      CMPB    (4)+,(6)+       ;TEST INCREMENT OF R4
10565  021766  020427  000001              CMP     %4,#1
10566  021772  001401                      BEQ     BR10
  (2)  021774  104000                      EMT                     ;WRONG INCREMENT OF R4
10567
10568  021776  012706  001000      BR10:   MOV     #1000,%6
10569  022002  124627  000000              CMPB    -(6),@HERE      ;TEST DECREMENT OF R6
10570  022006  022706  000776              CMP     #776,%6
10571  022012  001401                      BEQ     TS307
  (3)  022014  104000                      EMT                     ;WRONG DECREMENT OF R6,OR WRONG $TSTNM
10572                       ;*****************************************************************************
  (2)                       ;TEST 307       TEST TRANSFER OF .BYTE USING R6
  (3)                       ;*****************************************************************************
  (2)  022016               TS307:
10573  022016  012767  123456  177536      MOV     #123456,K5
10574  022024  012767  050505  177520      MOV     #050505,K1
10575  022032  012705  021552              MOV     #K1,%5          ;%5=(050505)K1
10576  022036  012706  021562              MOV     #K5,%6          ;%6=(123456)K5
10577  022042  112625                      MOVB    (6)+,(5)+       ;LOW .BYTE OF R6 TO R5
10578  022044  022767  050456  177500      CMP     #050456,K1
10579  022052  001401                      BEQ     BR11
  (2)  022054  104000                      EMT                     ;FALSE TRANSFER OF .BYTE
10580
10581  022056  012767  123456  177476  BR11:   MOV     #123456,K5
10582  022064  012767  050505  177460          MOV     #050505,K1
10583  022072  012705  021552                  MOV     #K1,%5          ;%5(050505)K1
10584  022076  012706  021564                  MOV     #K6,%6          ;%6(123456)K5
10585  022102  114625                          MOVB    -(6),(5)+       ;LOW .BYTE OF R6 TO R5 (DECREMENT)
10586  022104  026727  177442  050456          CMP     K1,#050456
```

```
022112  001401                          BEQ     BR12
022114  104000                          EMT                     ;FALSE R6 .BYTE TRANSFER

022116  012767  123456  177426  BR12:   MOV     #123456,K1
022124  012767  050505  177430          MOV     #050505,K5
022132  012705  021552                  MOV     #K1,#5          ;(123456)
022136  012706  021562                  MOV     #K5,#6          ;(050505)
022142  112526                          MOVB    (5)+,(6)+       ;LOW OF R5 TO LOW OF R6
022144  022767  050456  177410          CMP     #050456,K5
022152  001401                          BEQ     BR13
022154  104000                          EMT                     ;FALSE R6 .BYTE TRANSFER

022156  012767  123456  177366  BR13:   MOV     #123456,K1
022164  012767  050505  177370          MOV     #050505,K5
022172  012705  021553                  MOV     #K1+1,#5        ;123456
022176  012706  021562                  MOV     #K5,#6          ;050505
022202  112526                          MOVB    (5)+,(6)+       ;HIGH OF R5 TO LOW OF R6
022204  026727  177352  050647          CMP     K5,#050647
022212  001401                          BEQ     BR14
022214  104000                          EMT                     ;FALSE R6 .BYTE TRANSFER

022216  012767  123456  177326  BR14:   MOV     #123456,K1
022224  012767  050505  177330          MOV     #050505,K5
022232  012705  021553                  MOV     #K1+1,#5        ;R5-123456-ODD ADDRESS
022236  012706  021562                  MOV     #K5,#6          ;R6-050505--.EVEN ADDRESS
022242  112625                          MOVB    (6)+,(5)+       ;LOW OF R6 TO HIGH OF R5
022244  022767  042456  177300          CMP     #042456,K1
022252  001401                          BEQ     TS310
022254  104000                          EMT                     ;FAILED LOW OF 6 TO HIGH OF 5,OR WRONG $TSTNM
                                ;*******************************************************************************
                                ;TEST 310       TEST BYTE OPERATION WITH SEQUENTIAL ODD-EVEN ADDRESS
                                ;*******************************************************************************
022256                          TS310:
022256  126767  177304  177303          CMPB    K7,K7+1         ;SAME .WORD LOW TO HIGH
022264  001401                          BEQ     BR15
022266  104000                          EMT                     ;SHOULD COMPARE LOW TO HIGH

022270  126767  177273  177270  BR15:   CMPB    K7+1,K7         ;COMPARE ODD TO .EVEN SAME .WORD
022276  001401                          BEQ     BR16
022300  104000                          EMT                     ;ODD TO .EVEN .BYTE FAILURE

022302  126767  177263  177256  BR16:   CMPB    K10+1,K7        ;SEQUENTIAL .BYTES
022310  001401                          BEQ     BR17
022312  104000                          EMT                     ;ODD TO .EVEN FAILED

022314  126767  177250  177242  BR17:   CMPB    K10,K6
022322  001401                          BEQ     BR20
022324  104000                          EMT                     ;.EVEN TO EVEN FAILED
022326  126767  177235  177235  BR20:   CMPB    K7+1,K10+1
022334  001401                          BEQ     BR21
022336  104000                          EMT                     ;ODD TO ODD FAILED

022340  126767  177224  177223  BR21:   CMPB    K10,K10+1
022346  001001                          BNE     BR22
022350  104000                          EMT                     ;LOW TO HIGH IN SAME .WORD FAILED
```

```
10630  022352  126767  177213  177211  BR22:   CMPB    K10+1,K10+1
10631  022360  001401                          BEQ     BR23
  (2)  022362  104000                          EMT                         ;HIGH TO LOW IN SAME .WORD FAILED
10632
10633  022364  126767  177200  177175  BR23:   CMPB    K10,K7+1
10634  022372  001001                          BNE     TS311
  (3)  022374  104000                          EMT                         ;.EVEN TO ODD FAILED,OR WRONG $TSTNM
10635
10636
10637                                  ;*****************************************************************************
  (2)                                  ;TEST 311       TEST THAT DECREMENT R6 TO A VALUE LESS THAN 400 TRAPS
  (3)                                  ;*****************************************************************************
  (2)  022376                          TS311:
10638  022376  012706  000150                  MOV     #150,%6             ;R6 = 150
10639  022402  012767  022414  155374          MOV     #TDEC1,4            ;STACK OVERFLOW TRAP POINTER
10640  022410  005746                          TST     -(6)                ;WITH R6 = 150 SHOULD TRAP
10641  022412  104000                          EMT                         ;SHOULD HAVE TRAPPED,OR WRONG $TSTNM
10642  022414                       TDEC1:
10643
10644                                  ;*****************************************************************************
  (2)                                  ;TEST 312       TEST FOR DECREMENT OF R6 ON OVERFLOW TRAP
  (3)                                  ;*****************************************************************************
  (2)  022414                          TS312:
10645  022414  012706  000150                  MOV     #150,%6             ;R6 = 150
10646  022420  012767  022430  155356          MOV     #TDEC2,4            ;TRAP POINTER
10647  022426  005746                          TST     -(6)                ;WITH R6 = 150 SHOULD TRAP
10648  022430  020627  000142       TDEC2:     CMP     %6,#142             ;DID R6 DECREMENT
10649  022434  001401                          BEQ     TS313
  (3)  022436  104000                          EMT                         ;R6 NOT = 142,OR WRONG $TSTNM
10650
10651                                  ;*****************************************************************************
  (2)                                  ;TEST 313       TEST DIFFERENT TYPES OF OVERFLOW
  (3)                                  ;*****************************************************************************
  (2)  022440                          TS313:
10652  022440  012706  000150                  MOV     #150,%6
10653  022444  005067  155476                  CLR     146                 ;STATUS WORD OF LOC 10
10654  022450  012767  022460  155326          MOV     #TDEC3,4            ;RETURN TO LOC 4
10655  022456  005246                          INC     -(6)
10656  022460  005767  155462       TDEC3:     TST     146
10657  022464  001001                          BNE     1$
  (2)  022466  104000                          EMT                         ;INCREMENT OPERATION NOT INHIBITED
10658  022470  012705  001000       1$:        MOV     #1000,%5
10659  022474  012706  000400                  MOV     #400,%6
10660  022500  012767  022512  155276          MOV     #TDEC4,4
10661  022506  124645                          CMPB    -(6),-(5)
10662  022510  104000                          EMT                         ;STACK = 400 AND DECREMENTED, SHOULD TRAP
10663  022512  012706  000400       TDEC4:     MOV     #400,%6
10664  022516  012767  022530  155260          MOV     #TDEC7,4
10665  022524  134546                          BITB    -(5),-(6)
10666  022526                       TDEC6:
  (2)  022526  104000                          EMT                         ;NO STACK OVERFLOW,OR WRONG $TSTNM
10667  022530                       TDEC7:
10677
10678                                  ;*****************************************************************************
  (3)                                  ;TEST 314       TEST THAT AN 77 CAUSES AN OVERFLOW TRAP
  (4)                                  ;*****************************************************************************
```

```
   (3)  022530                                  TS314:
   (1)  022530  012706  000400                          MOV     #400,%6         ;SET UP STACK TO OVERFLOW
   (1)  022534  012767  022552  155246                  MOV     #VDEC2,10       ;SET UP 77 VECTOR
   (1)  022542  012767  022556  155234                  MOV     #VDEC,4 ;SET UP OVERFLOW VECTOR
   (1)  022550  000077                                  77                      ;THIS TRAP SHOULD CAUSE OVERFLOW
   (1)  022552  000167  157530                VDEC2:    JMP     ERROR1          ;USE JUMP TO GET TO ERROR BECAUSE UNSURE WHAT EMT WILL D
   (1)  022556  012767  021530  155224        VDEC:     MOV     #T010,10        ;RESTORE VECTOR
10679                                          ;*****************************************************************************
   (3)                                         ;TEST 315       TEST THAT AN IOT CAUSES AN OVERFLOW TRAP
   (4)                                         ;*****************************************************************************
   (3)  022564                                 TS315:
   (1)  022564  012706  000400                          MOV     #400,%6         ;SET UP STACK TO OVERFLOW
   (1)  022570  012767  022606  155222                  MOV     #VDEC4,20       ;SET UP IOT VECTOR
   (1)  022576  012767  022612  155200                  MOV     #VDEC3,4        ;SET UP OVERFLOW VECTOR
   (1)  022604  000004                                  IOT                     ;THIS TRAP SHOULD CAUSE OVERFLOW
   (1)  022606  000167  157474                VDEC4:    JMP     ERROR1          ;USE JUMP TO GET TO ERROR BECAUSE UNSURE WHAT EMT WILL D
   (1)  022612  012767  021534  155200        VDEC3:    MOV     #T020,20        ;RESTORE VECTOR
10680
10681                                          ;*****************************************************************************
   (2)                                         ;TEST 316       TEST THAT AN EMT CAUSES AN OVERFLOW TRAP (CHECK OF YELLOW ZONE)
   (3)                                         ;*****************************************************************************
   (2)  022620                                 TS316:
10682  022620  012706  000400                          MOV     #400,%6         ;SET UP STACK TO OVERFLOW
10683  022624  012767  022642  155176                  MOV     #VDEC6,30       ;SET UP INST VECTOR
10684  022632  012767  022646  155144                  MOV     #VDEC5,4        ;SET UP OVERFLOW BECTOR
10685  022640  104000                                  EMT                     ;THIS TRAP SHOULD CAUSE OVERFLOW
10686  022642  000167  157440                VDEC6:    JMP     ERROR1          ;USE JUMP TO GET TO ERROR BECAUSE UNSURE WHAT EMT WILL D
10687  022646  012767  002306  155154        VDEC5:    MOV     #ERROR1,30      ;RESTORE VECTOR
10688
10689                                          ;*****************************************************************************
   (3)                                         ;TEST 317       TEST THAT AN TRAP CAUSES AN OVERFLOW TRAP
   (4)                                         ;*****************************************************************************
   (3)  022654                                 TS317:
   (1)  022654  012706  000400                          MOV     #400,%6         ;SET UP STACK TO OVERFLOW
   (1)  022660  012767  022676  155146                  MOV     #VDEC8,34       ;SET UP TRAP VECTOR
   (1)  022666  012767  022702  155110                  MOV     #VDEC7,4        ;SET UP OVERFLOW VECTOR
   (1)  022674  104400                                  TRAP                    ;THIS TRAP SHOULD CAUSE OVERFLOW
   (1)  022676  000167  157404                VDEC8:    JMP     ERROR1          ;USE JUMP TO GET TO ERROR BECAUSE UNSURE WHAT EMT WILL D
   (1)  022702  012767  021540  155124        VDEC7:    MOV     #T034,34        ;RESTORE VECTOR
10690                                          ;*****************************************************************************
   (3)                                         ;TEST 320       TEST THAT AN TRT CAUSES AN OVERFLOW TRAP
   (4)                                         ;*****************************************************************************
   (3)  022710                                 TS320:
   (1)  022710  012706  000400                          MOV     #400,%6         ;SET UP STACK TO OVERFLOW
   (1)  022714  012767  022732  155072                  MOV     #VDEC10,14      ;SET UP TRT VECTOR
   (1)  022722  012767  022736  155054                  MOV     #VDEC9,4        ;SET UP OVERFLOW VECTOR
   (1)  022730  000003                                  TRT                     ;THIS TRAP SHOULD CAUSE OVERFLOW
   (1)  022732  000167  157350                VDEC10:   JMP     ERROR1          ;USE JUMP TO GET TO ERROR BECAUSE UNSURE WHAT EMT WILL D
   (1)  022736  012767  021532  155050        VDEC9:    MOV     #T014,14        ;RESTORE VECTOR
10691                                          ;*****************************************************************************
   (3)                                         ;TEST 321       TEST THAT AN ILLA CAUSES AN OVERFLOW TRAP
   (4)                                         ;*****************************************************************************
   (3)  022744                                 TS321:
   (1)  022744  012706  000400                          MOV     #400,%6         ;SET UP STACK TO OVERFLOW
   (1)  022750  012767  022766  155026                  MOV     #VDEC11,4       ;SET UP ILLA VECTOR
   (1)  022756  012767  022772  155020                  MOV     #VDEC12,4       ;SET UP OVERFLOW VECTOR
```

```
  (1)  022764  004700                          ILLA                    ;THIS TRAP SHOULD CAUSE OVERFLOW
  (1)  022766  000167  157314          VDEC11: JMP     ERROR1          ;USE JUMP TO GET TO ERROR BECAUSE UNSURE WHAT EMT WILL D
  (1)  022772  012767  021526  155004  VDEC12: MOV     #T04,4  ;RESTORE VECTOR
10692  023000  020627  000370                  CMP     #6,#370         ;STACK PUSHED FOUR WORDS?
10693  023004  001401                          BEQ     TS322
  (3)  023006  104000                          EMT                     ;TRAP OVERFLOW DID NOT OCCUR
10694                                  ;*******************************************************************************
  (3)                                  ;TEST 322       TEST THAT AN ILLB CAUSES AN OVERFLOW TRAP
  (4)                                  ;*******************************************************************************
  (3)  023010                          TS322:
  (1)  023010  012706  000400                  MOV     #400,#6         ;SET UP STACK TO OVERFLOW
  (1)  023014  012767  023032  154762          MOV     #VDEC13,4       ;SET UP ILLB VECTOR
  (1)  023022  012767  023036  154754          MOV     #VDEC14,4       ;SET UP OVERFLOW VECTOR
  (1)  023030  000100                          ILLB                    ;THIS TRAP SHOULD CAUSE OVERFLOW
  (1)  023032  000167  157250          VDEC13: JMP     ERROR1          ;USE JUMP TO GET TO ERROR BECAUSE UNSURE WHAT EMT WILL D
  (1)  023036  012767  021526  154740  VDEC14: MOV     #T04,4  ;RESTORE VECTOR
10695
10696                                  ;*******************************************************************************
  (2)                                  ;TEST 323       TEST FOR FALSE OVERFLOW TRAP
  (3)                                  ;*******************************************************************************
  (2)  023044                          TS323:
10697
10698  023044  012767  023112  154732          MOV     #FOVER,4        ;SET UP OVERFLOW POINTER
10699  023052  012706  001002                  MOV     #1002,#6
10700  023056  005746                          TST     -(6)            ;SHOULD NOT OVERFLOW
10701  023060  012706  002002                  MOV     #2002,#6
10702  023064  005746                          TST     -(6)            ;SHOULD NOT OVERFLOW
10703  023066  012706  004002                  MOV     #4002,#6
10704  023072  005746                          TST     -(6)            ;SHOULD NOT OVERFLOW
10705  023074  012706  010002                  MOV     #10002,#6
10706  023100  005746                          TST     -(6)
10707  023102  012706  020000                  MOV     #20000,#6       ;SHOULD NOT OVERFLOW
10708  023106  005746                          TST     -(6)
10709  023110  000401                          BR      STP
  (1)  023112                          FOVER:
  (2)  023112  104000                          EMT                     ;IT OVERFLOWED,OR WRONG $TSTNM
10710  023114  012767  021526  154662  STP:    MOV     #T04,4
10711  023122  005067  154660                  CLR     6
10712                                  ;*******************************************************************************
  (2)                                  ;TEST 324       TEST THAT BIT 4 PSW WILL CAUSE A TRAP TO 14
  (3)                                  ;*******************************************************************************
  (2)  023126                          TS324:
10713  023126  012706  001000                  MOV     #STBOT,SP
10714  023132  012767  023156  154654          MOV     #RETAT,RTRAP4   ;SET UP TO TRAP TO 14
10715  023140  012746  000020                  MOV     #20,-(SP)       ;PUSH T BIT
10716  023144  012746  023152                  MOV     #.+6,-(SP)      ;PUSH PC
10717  023150  000002                          RTI                     ;SET T BIT
10718  023152  000240                          NOP                     ;TRAP HERE
10719  023154  104000                          EMT                     ;TRACE BIT DID NOT TRAP!,OR WRONG $TESTN
10720  023156                          RETAT:
10721                                  ;*******************************************************************************
  (2)                                  ;TEST 325       TEST  STACK POINTER DECREMENTS
  (3)                                  ;*******************************************************************************
  (2)  023156                          TS325:
10722  023156  012706  001000                  MOV     #STBOT,SP
10723  023162  012767  023206  154624          MOV     #RETBT,RTRAP4
```

```
10724  023170  012746  000020                  MOV     #20,-(SP)       ;PUSH T BIT
10725  023174  012746  023202                  MOV     #.+6,-(SP)      ;PUSH PC
10726  023200  000002                          RTI                     ;SET T BIT
10727  023202  000240                          NOP                     ;TRAP HERE
10728  023204  104000                          EMT                     ;TRACE BIT DID NOT TRAP!
10729  023206  020627  000774          RETBT:  CMP     SP,#STBOT-4
10730  023212  001401                          BEQ     TS326
  (3)  023214  104000                          EMT                     ;STACK POINTER WAS NOT PUSHED BY TRAP,OR WRONG $TESTN
10731                                  ;*****************************************************************************
  (2)                                   ;TEST 326       TEST  FOR PROPER PC ON STACK
  (3)                                   ;*****************************************************************************
  (2)  023216                           TS326:
10732  023216  012706  001000                  MOV     #STBOT,SP
10733  023222  012767  023242  154564          MOV     #RETCT,RTRAP4
10734  023230  012746  000020                  MOV     #20,-(SP)       ;PUSH T BIT
10735  023234  012746  023242                  MOV     #.+6,-(SP)      ;PUSH PC
10736  023240  000002                          RTI                     ;SET T BIT
10737                                                                  ;TRAP HERE
10738  023242  022767  023242  155524  RETCT:  CMP     #.,STBOT-4
10739  023250  001401                          BEQ     TS327
  (3)  023252  104000                          EMT                     ;CORRECT PC WAS NOT SAVED ON STACK,OR WRONG $TESTN
10740
10741
10742                                   ;*****************************************************************************
  (2)                                   ;TEST 327       TEST  THAT RTT POPS T- BIT
  (3)                                   ;*****************************************************************************
  (2)  023254                           TS327:
10743
10744  023254  012706  001000                  MOV     #STBOT,SP
10745  023260  005001                          CLR     R1              ;CLEAR R1
10746  023262  012746  000020                  MOV     #20,-(SP)
10747  023266  012746  023302                  MOV     #RTT1,-(SP)
10748  023272  012767  023310  154514          MOV     #RTT2,14
10749  023300  000006                          RTT
10750  023302  000240                  RTT1:   NOP
10751  023304  001401                          BEQ     TS330
  (3)  023306  104000                          EMT                     ;T-BIT DID NOT TRAP,OR WRONG $TESTN
10752
10753  023310                          RTT2:
10754                                   ;*****************************************************************************
  (2)                                   ;TEST 330       TEST  THAT RTT ALLOWS ONE INST. BEFORE TRAP
  (3)                                   ;*****************************************************************************
  (2)  023310                           TS330:
10755  023310  012705  177777                  MOV     #177777,@#5
10756  023314  012706  001000          RTT5:   MOV     #STBOT,SP
10757  023320  012746  000020                  MOV     #20,-(SP)
10758  023324  012746  023342                  MOV     #RTT3,-(SP)
10759  023330  012767  023352  154456          MOV     #RTT4,14
10760  023336  005001                          CLR     R1              ;CLEAR R0
10761  023340  000006                          RTT                     ;SET T-BIT
10762  023342  005201                  RTT3:   INC     R1
10763  023344  005205                          INC     @#5
10764  023346  001762                          BEQ     RTT5            ;DO THIS TEST NO MORE THAN 2 TIMES
10765  023350  104000                          EMT                     ;DID NOT TRAP
10766  023352  005301                  RTT4:   DEC     R1              ;SEE IF RTT ALLOWS 1 INST.
10767  023354  001403                          BEQ     RTT6
```

```
023356  005205                          INC     #5                  ;DO THIS TEST NO MORE THAN TWO TIMES
023360  001755                          BEQ     RTT5
023362  104000                          EMT                         ;RTT DID NOT ALLOW 1 INST.,OR WRONG $TESTN
023364                          RTT6:
                                ;*******************************************************************************
                                ;TEST 331       TEST  THAT RTI DOES NOT ALLOW 1 INST.
                                ;*******************************************************************************
023364                          TS331:
023364  012706  001000                  MOV     #STBOT,SP
023370  012746  000020                  MOV     #20,-(SP)
023374  012746  023412                  MOV     #RTI1,-(SP)
023400  012767  023416  154406          MOV     #RTI2,14
023406  005001                          CLR     R1
023410  000002                          RTI                         ;SET T-BIT
023412  005201                  RTI1:   INC     R1                  ;RTI SHOULD NOT ALLOW THIS
023414  104000                          EMT                         ;T- BIT DID NOT CAUSE TRAP
023416  005701                  RTI2:   TST     R1
                                                                    ;RTI SHOULD NOT ALLOW 1 INST. BEFORE TRAP
023420  001401                          BEQ     TS332
023422  104000                          EMT                         ;RTI DID ALLOW 1 INST. BEFORE TRAP,OR WRONG $TESTN

                                ;*******************************************************************************
                                ;TEST 332       TEST TRAP ON TRAP THAT TRACE BIT TRAPS ARE INHIBITED ON TRAP INST
                                ;*******************************************************************************
023424                          TS332:

023424  012706  001000                  MOV     #STBOT,#6
023430  012767  023470  154356          MOV     #TRACE,14           ;TRACE TRAP
023436  005027  000016                  CLR     #16
023442  005027  000022                  CLR     #22
023446  012767  023474  154344          MOV     #TONT1,20           ;IOT TRAP
023454  012746  000020                  MOV     #20,-(SP)           ;PUSH T BIT
023460  012746  023466                  MOV     #.+6,-(SP)          ;PUSH PC
023464  000006                          RTT
023466  000004                          IOT                         ;TRAP, NEW CC HAVE TRACE RESET
023470                          TRACE:
023470  104000                          EMT                         ;TRACE TRAP WAS NOT INHIBITED
023472                          BR70:
023472  104000                          EMT                         ;WRONG TSTNM,OR WRONG $TSTNM
023474  012767  000016  154312  TONT1:  MOV     #16,14
023502  012767  000022  154310          MOV     #22,20
```

```
10800                                   ;*****************************************************************************
  (2)                                   ;TEST 333       TEST THAT THE TRACE BIT IS SAVED IN THE STACK
  (3)                                   ;*****************************************************************************
  (2)  023510                           TS333:
10801  023510  012706  001000                      MOV     #STBOT,#6       ;SET UP STACK POINTER
10802  023514  012767  023540  154272              MOV     #TRC1,14        ;TRACE TRAP RETURN
10803  023522  005067  154270                      CLR     16
10804  023526  012746  000020                      MOV     #20,-(SP)       ;SET THE T BIT
10805  023532  012746  023540                      MOV     #TRC1,-(SP)
10806  023536  000002                              RTI
10807  023540  036727  155232  000020  TRC1:       BIT     STBOT-2,#20     ;CHECK FOR T BIT ON STACK
10808  023546  001001                              BNE     STP3D
  (1)  023550                           STP3:
  (2)  023550  104000                              EMT                     ;T BIT NOT SAVED ON THE STACK,OR WRONG $TSTNM
10809  023552  012767  021532  154234  STP3D:     MOV     #TO14,14
10810
```

```
                        ;*************************************************************************
                        ;THIS ROUTINE TESTS THAT NO LEGAL ADDRESS TRAPS AND THAT AN ILLEGAL
                        ;ADDRESS TRAPS TO LOCATION 4.  THIS WILL RUN ON 30K SYSTEM.  BUT IF
                        ;SWITCH REGISTER BIT 1=0, THEN THE MEMORY FROM 28K-30K IS NOT LOOKED
                        ;AT, SINCE IT MAY HAVE I/O DEVICES.  IF SWR BIT 1=1, THEN THAT AREA IS
                        ;CHECKED.  (IT SHOULD EITHER ALL TRAP OR ALL NOT TRAP).  LOC 160000
                        ;IS NO LONGER GUARANTEED TO TRAP, SINCE IT MAY CONTAIN MEMORY. LOCATION
                        ;177700 (THE UNIBUS ADDRESS FOR R0 ON OLDER SYSTEMS) IS USED FOR FORCING
                        ;A TIMEOUT IN THE EVENT THAT THERE WAS NO TIMEOUT FROM 0K-28K OR 30K.
                        ;THIS ROUTINE TESTS MEMORY UNTIL IT DOES A NXM STOP
                        ;*************************************************************************
                        ;TEST 334       TEST NON-EXISTENT ADDRESS TRAPS
                        ;*************************************************************************
                        TS334:
                        1$:     CLR     R0              ;
                                CLR     6               ;
                                MOV     #ATRAP,4        ;SET UP ADDRESS TRAP ENTRANCE
                                MOV     #STBOT,SP       ;SET STACK POINTER
                        NOR:    TSTB    (0)+            ;IF OUTSIDE OF CORE, TRAP TO 4
                                CMP     R0,(PC)+        ;IS POINTER INSIDE 28K (30K) CORE
                        HICORE: .WORD   160000          ;MAY BE CHANGED TO 170000 IF 30K
                                BLO     NOR             ;TEST THE REST OF CORE
                                MOV     #ROTRAP,@#4     ;SET UP NEW VECTOR POINTER
                                TSTB    @#177700        ;SHOULD CAUSE A TRAP
                        TRPADR:
                                EMT                     ;SHOULD HAVE TRAPED
                        ;TRAP TO HERE IF FORCING TRAP BY TESTING 177700
                        ROTRAP: MFPS    STATUS
                                TST     STATUS          ;TEST PSW
                                BEQ     1$
                                EMT                     ;NEW PSW SHOULD HAVE BEEN ZERO
                        1$:     CMP     STBOT-4,#TRPADR ;TEST OLD PC AT STACK
                                BEQ     TRAPB
                                EMT                     ;OLD PC WAS NOT SAVED
                        ;RETURN HERE ON AN ADDRESS TRAP FROM MEMORY BELOW 28K (OR 30K)
                        ATRAP:  DEC     R0              ;
                                MOV     R0,CORH         ;MOVE THE FIRST NXM LOCATION IN CORH
                        ;THIS ROUTINE DOES NXM TRAPS UNTIL IT FINDS AN EXISTENT MEMORY LOCATION
                                MOV     @#HICORE,R0     ;SET UP THE HIGHEST MEM LOCATION
                                DEC     R0              ;MAKE 1 LESS THAN THE HIGHEST CORE BOUNDARY
                                BR      NOSUB           ;DON'T SUBTRACT 1K FIRST TIME
                        CTRAP:  SUB     #1000,R0        ;SUBTRACT 1K OCTAL BYTE FROM ADDRESS
                                                        ;TO SPEED UP TESTING
                        NOSUB:  MOV     #BTRAP,4        ;SET UP THE VECTOR
                                MOV     #STBOT,SP
                                TST     (R0)            ;DOES THIS MEMORY EXIST?
                                                        ;IF NXM, TRAP TO BTRAP
                        DTRAP1: CMP     R0,(PC)+        ;IF EXISTS, IS THIS THE SAME TRAP THAT CAUSED
                                                        ;TRAP TO ATRAP
                        CORH:   .WORD   0
                                BLOS    TRAPB
                                EMT                     ;CONTENTS OF R0 SHOULD BE LESS THAN OR EQUAL  TO CORH
                                                        ;IF THIS COMPARISON FAILS IT MEANS
                                                        ;THAT SOME LEGAL ADDRESS TRAPPED, OR
                                                        ;THAT AN ILLEGAL ADDRESS DID NOT TRAP
                        BTRAP:  MFPS    STATUS
```

```
10861  023724  005767  154046                  TST     STATUS
10862  023730  001401                          BEQ     1$
  (2)  023732  104000                          EMT                             ;NEW PSW SHOULD HAVE BEEN ZERO
10863  023734  026727  155034  023710  1$:     CMP     STBOT-4,#DTRAP1 ;CHECK IF TRAP PC IS OK
10864  023742  001752                          BEQ     CTRAP
  (1)  023744                          AUTO1:
  (2)  023744  104000                          EMT                             ;OLD PC WAS NOT SAVED OR WRONG $TESTN
10865  023746  012767  021526  154030  TRAPB:  MOV     #T04,4          ;RESET TRAP CATCHER
10866  023754  005067  154026                  CLR     6               ;RESET TRAP CATCHER
10867
10868
10869                                  ;THIS ROUTINE WILL FIGURE OUT IF YOU HAVE A DL11W
10870  023760  012706  001000                  MOV     #STBOT,SP       ;SET UP THE STACK POINTER
10871  023764  012767  024000  154012          MOV     #NODL,4         ;SET UP THE TRAP VECTOR
10872  023772  005767  153566                  TST     TTCSR           ;TEST THE PUNCH STATUS REGISTER
10873  023776  000405                          BR      DL11W
10874  024000  012767  021526  153776  NODL:   MOV     #T04,4
10875  024006  000167  100624                  JMP     SLU1ST          ;IF NO SLU FIND OUT WHY IN SLU TEST
10876  024012  012767  021526  153764  DL11W:  MOV     #T04,4
10877
10878                                  ;******************************************************************************
  (2)                                 ;TEST 335       TEST THAT A TTY INTERRUPT CAUSES AN OVERFLOW TRAP
  (3)                                 ;******************************************************************************
  (2)  024020                          TS335:
10879  024020  012767  000340  153750          MOV     #340,STATUS     ;LOCK OUT INTERRUPT
10880  024026  012706  000400                  MOV     #400,%6         ;SET UP STACK TO OVERFLOW
10881  024032  012767  024074  153744          MOV     #TDEC77,4       ;SET UP OVERFLOW TRAP
10882  024040  016767  154020  001562          MOV     64,     TEMP1   ;SAVE CONTENTS OF INTERRUPT VECTOR
10883  024046  012767  024072  154010          MOV     #TDEC8,64       ;SET UP INTERRUPT VECTOR
10884  024054  012767  000100  153502          MOV     #100,TTCSR      ;SET INTERRUPT ENABLE
10885  024062  005067  153710                  CLR     STATUS          ;ALLOW INTERRUPT TO OCCUR
10886  024066  000167  100544                  JMP     SLU1ST          ;NO INTERRUPT OCCURRED SO GO TO SLU TEST
10887                                                                  ;TO FIND OUT WHY ADD REPORT PROPER ERROR
10888  024072                          TDEC8:
  (2)  024072  104000                          EMT                             ;OVERFLOW TRAP DID NOT OCCUR
10889  024074  005067  153464          TDEC77: CLR     TTCSR           ;CLEAR INTERRUPT ENABLE
10890  024100  012767  021526  153676          MOV     #T04,4
10891  024106  005067  153674                  CLR     6
10892  024112  016767  001512  153744          MOV     TEMP1,  64      ;RESTORE CONTENTS OF INTERRUPT VECTOR
10893                                  ;******************************************************************************
  (2)                                 ;TEST 336       TEST THAT A PENDING INTERRUPT OCCURS BEFORE TRAP
  (3)                                 ;******************************************************************************
  (2)  024120                          TS336:
10894  024120  012706  001000                  MOV     #STBOT,%6
10895  024124  012767  000340  153644          MOV     #340,STATUS     ;SET TO A HIGH PRIORITY LEVEL
10896  024132  016767  153726  001470          MOV     64,     TEMP1   ;SAVE CONTENTS OF INTERRUPT VECTOR
10897  024140  012767  024204  153716          MOV     #TR0,64
10898  024146  012767  000100  153410          MOV     #100,TTCSR      ;INTERRUPT FOR TTY PUNCH/PRINTER
10899  024154  012767  024206  153652          MOV     #BR71,34        ;TRAP VECTOR
10900  024162  012767  024210  153674          MOV     #TR2,64         ;TTY VECTOR
10901  024170  012767  000340  153640          MOV     #340,36         ;IF TRAP TRAPS, MOVE 340 TO PRIORITY
10902  024176  005067  153574                  CLR     STATUS          ;SHOULD INTERRUPT AT END OF CLR INST
10903  024202  104400                          TRAP                    ;TTY INTERRUPT SHOULD OVERRIDE TRAP
10904  024204                          TR0:
  (2)  024204  104000                          EMT                             ;TTY SHOULDN'T HAVE INTERRUPTED
10905  024206                          BR71:
```

```
  (2)  024206  104000                          EMT                         ;TRAP OCCURRED FIRST
10906  024210  005067  153622            TR2:  CLR     36
10907  024214  016767  001410  153642        MOV     TEMP1,  64            ;RESTORE CONTENTS OF INTERRUPT VECTOR
10908  024222  042767  000100  153334        BIC     #100,TTCSR
10909                                ;*************************************************************************************
  (2)                                ;TEST 337       TEST THAT A PENDING INTERRUPT, INTERRUPTS BETWEEN TRAPS
  (3)                                ;*************************************************************************************
  (2)  024230                        TS337:
10910  024230  012706  001000            MOV     #STBOT,%6
10911  024234  012767  000340  153534    MOV     #340,STATUS
10912  024242  012767  000100  153314    MOV     #100,TTCSR
10913  024250  012767  024316  153556    MOV     #TR3,34               ;TRAP
10914  024256  016767  153602  001344    MOV     64,     TEMP1         ;SAVE CONTENTS OF INTERRUPT VECTOR
10915  024264  012767  024322  153572    MOV     #TR4,64               ;TTY OUTPUT
10916  024272  012767  000340  153566    MOV     #340,66               ;TTY OUTPUT PRIORITY
10917  024300  012767  024320  153512    MOV     #TR5,20               ;IOT
10918  024306  012767  000340  153506    MOV     #340,22               ;IOT PRIORITY
10919  024314  104400                    TRAP                          ;THE ACT OF TRAPPING LOWER PRIORITY
10920  024316  000004                TR3:  IOT                           ;INTERRUPT SHOULD OCCUR IN PLACE OF IOT TRAP
10921  024320                        TR5:
  (2)  024320  104000                    EMT                           ;NO INTERRUPT BETWEEN TRAPS,OR WRONG $TSTNM
10922  024322  005067  153474        TR4:  CLR     22                    ;CLR IOT PRIORITY
10923  024326  005067  153534            CLR     66
10924  024332  012767  021540  153474    MOV     #T034,34
10925  024340  016767  001264  153516    MOV     TEMP1,  64            ;RESTORE CONTENTS OF INTERRUPT VECTOR
10926  024346  012767  000022  153444    MOV     #22,20
10927  024354  042767  000100  153202    BIC     #100,TTCSR            ;CLEAR IE BIT IN SLU1 XMIT CSR
10928
10929                                ;*************************************************************************************
  (2)                                ;TEST 340       TEST THAT "RESET" GOES TO OUTSIDE WORLD
  (3)                                ;*************************************************************************************
  (2)  024362                        TS340:
10930  024362  032737  000001  001020        BIT     #1, @#$ENV            ;ARE WE RUNNING UNDER APT
  (1)  024370  001403                        BEQ     70$                   ;IF NO THEN DO TEST
  (1)  024372  005737  001006                TST     @#$PASS               ;IS THIS FIRST PASS
  (2)  024376  001013                        BNE     TS341                 ;IF NO THEN SHIP TO NEXT TEST
  (1)  024400                        70$:
10931  024400  016700  153156            MOV     TKB,R0                ;MAKE SURE RECEIVER DONE IS CLEAR
10932  024404  012767  000100  153146    MOV     #100,TRCSR            ;SET INTERRUPT ENABLE
10933  024412  000005                    RESET                         ;SHOULD CLEAR INTERRUPT ENABLE
10934  024414  032767  000100  153136    BIT     #100,TRCSR            ;TEST FOR CLEAR
10935  024422  001401                    BEQ     TS341
  (3)  024424  104000                    EMT                           ;RESET FAILED TO CLEAR TRCSR,OR WRONG $TSTNM
10936  024426                        NODL2:
10937                                ;*************************************************************************************
  (2)                                ;TEST 341       TEST THAT RESET HAS NO EFFECT ON THE TRACE TRAP
  (3)                                ;*************************************************************************************
  (2)  024426                        TS341:
10938  024426  032737  000001  001020        BIT     #1, @#$ENV            ;ARE WE RUNNING UNDER APT
  (1)  024434  001403                        BEQ     70$                   ;IF NO THEN DO TEST
  (1)  024436  005737  001006                TST     @#$PASS               ;IS THIS FIRST PASS
  (2)  024442  001024                        BNE     TS342                 ;IF NO THEN SHIP TO NEXT TEST
  (1)  024444                        70$:
10939  024444  012706  001000            MOV     #STBOT,%6             ;SET STACK
10940  024450  012767  024476  153336    MOV     #RESET2,14            ;SET UP TRACE VECTOR
10941  024456  012746  000020            MOV     #20,-(R6)             ;SET THE T-BIT ON STACK
```

```
10942  024462  012746  024470                  MOV     #1$,-(R6)       ;MOVE NEW PC ON STACK
10943  024466  000006                          RTT
10944  024470  000005                  1$:     RESET                   ;SHOULD HAVE NO EFFECT
10945  024472  000005                          RESET                   ;NO EFFECT
10946  024474                          RESET3:
  (2)  024474  104000                          EMT                     ;TRACE TRAP FAILED,OR WRONG $TSTNM
10947  024476  005067  153274          RESET2: CLR     STATUS          ;CLEAR TRACK
10948  024502  005067  153310                  CLR     16              ;TRACE STATUS
10949  024506  012767  021532  153300          MOV     #TO14,14
10950  024514                          SKTST2:
10951
10952                                  ;*****************************************************************************
  (2)                                 ;TEST 342       TEST THE 'WAIT' INSTRUCTION
  (3)                                 ;*****************************************************************************
  (2)  024514                         TS342:
10953  024514  032737  000040  000052          BIT     #BIT5,@#52              ;ARE WE UNDER UFD ?
10954  024522  001402                          BEQ     5$                      ;NO,THEN EXECUTE THIS TEST
10955  024524  000167  000154                  JMP     NXTST                   ;YES,THEN GO TO THE NEXT TEST
10956  024530  122767  000001  154262  5$:     CMPB    #APTENV,$ENV            ;RUNING IN APT MODE?
10957  024536  001003                          BNE     1$                      ;IF NOT, DO THIS TEST
10958  024540  005767  154242                  TST     $PASS                   ;IS THIS THE FIRST PASS?
10959  024544  001051                          BNE     STP4E                   ;IF NOT FIRST PASS, SKIP TEST
10960  024546                          1$:
10961  024546  042767  000100  153010          BIC     #100,TTCSR      ;CLEAR INTERRUPT ENABLE
10962  024554  012706  001000                  MOV     #STBOT,SP       ;SET UP THE STACK
10963  024560  016767  153300  001042          MOV     64,     TEMP1   ;SAVE CONTENTS OF INTERRUPT VECTOR
10964  024566  012767  024646  153270          MOV     #WATE,64        ;SET UP THE INTERRUPT VECTOR
10965  024574  005067  153266                  CLR     66
10966  024600  105767  152760          WATE1:  TSTB    TTCSR           ;WAIT FOR READY
10967  024604  100375                          BPL     WATE1           ;TO BE UP
10968  024606  012767  000015  152752          MOV     #15,TPB         ;DO A CARRIAGE RETURN
10969  024614  105767  152744          WATE2:  TSTB    TTCSR           ;WAIT FOR READY TO COME UP
10970  024620  100375                          BPL     WATE2
10971  024622  012767  000015  152736          MOV     #15,TPB         ;DO ANOTHER CARRIAGE RETURN
10972  024630  052767  000100  152726          BIS     #100,TTCSR      ;SET THE INTERRUPT ENABLE
10973  024636  005067  153134                  CLR     STATUS          ;CLEAR THE PSW
10974  024642  000001                  WATE3:  WAIT                    ;WAIT FOR THE INTERRUPT
10975  024644  104000                          EMT                     ;WAIT INSTRUCTION DID NOT LOOP
10976  024646  005767  153124          WATE:   TST     STATUS  ;IS THE PSW CORRECT?
10977  024652  001401                          BEQ     1$
  (2)  024654  104000                          EMT                     ;NEW PSW SHOULD HAVE BEEN ZERO
10978  024656  026727  154112  024644  1$:     CMP     STBOT-4,@#WATE3+2       ;IS THE OLD PC SAVED
10979  024664  001401                          BEQ     STP4E
  (1)  024666                          STP4:
  (2)  024666  104000                          EMT                     ;OLD PC WAS NOT SAVED OR WRONG $TESTN
10980  024670  016767  000734  153166  STP4E:  MOV     TEMP1,  64      ;RESTORE CONTENTS OF INTERRUPT VECTOR
10981  024676  042767  000100  152660          BIC     #100,TTCSR      ;CLEAR IE BIT IN SLU1 XMIT CSR
10982                                  ;*****************************************************************************
  (2)                                 ;TEST 343       TEST THAT USING REGISTER ADDR (177700) CAUSES TIME OUT.
  (3)                                 ;*****************************************************************************
  (2)  024704                         TS343:
10983                                           ;
10984                                           ;REGISTER ADDRESS (177700-177717) CAUSE TIME OUT WHEN USED
10985                                           ;AS PROGRAM ADDRESS BY THE CPU.
10986                                           ;
10987  024704  012706  001000          NXTST:  MOV     #STBOT,SP       ;SET  STACK POINTER
```

```
10988  024710  012767  024724  153066          MOV     #RETR1,RTRAP5  ;SET TRAP RETURN ADDR
10989  024716  005237  177700          PCN1:   INC     @#177700       ;BAD ADDR REFERENCE, TRAP TO 4
10990  024722  104000                          EMT                    ;REFERENCING 177700 DID NOT CAUSE TIME OUT
10991  024724  022767  024722  154042  RETR1:  CMP     #PCN1+4,STBOT-4 ;PROPER PC STORED ON STACK?
10992  024732  001401                          BEQ     TS344
  (3)  024734  104000                          EMT                    ;OLD PC WAS NOT SAVED IN STACK
10993
10994                                          ;*************************************************************
10995                                          ;ODD ADDRESS USED BY A "WORD" INSTRUCTION SHOULD NOT
10996                                          ;CAUSE A TRAP, BUT THE LOW ORDER ADDRESS BIT WOULD BE IGNORED.
10997                                  ;*****************************************************************************
  (2)                                 ;TEST 344       TEST ODD ADDRESS TRAP IS NOT IMPLEMENTED.
  (3)                                 ;*****************************************************************************
  (2)  024736                         TS344:
10998                                          ;
10999  024736  012737  024762  000004          MOV     #RETR2,@#RTRAP5 ;SET TRAP RETURN ADDR
11000  024744  005037  000000                  CLR     @#0            ;PUT ALL 0 IN LOC 0
11001  024750  005337  000001                  DEC     @#1            ;DECREMENT ODD ADDRESS, SHOULD NOT TRAP
11002  024754  022737  177777  000000          CMP     #-1,@#0        ;WORD LOC 0 HAS ALL ONES?
11003  024762                          RETR2:
  (2)  024762  001401                          BEQ     TS345
  (3)  024764  104000                          EMT                    ;LOC 0 DID NOT STORE -1,OR ODD ADDR REFERENCE CAUSE TRAP
11004
11005                                          ;*************************************************************
11006                                          ;
11007                                          ;USING ADDRESS 177700 IN MODE 2, CAUSES BUS ERROR, BUT
11008                                          ;THE REGISTER IN USE WILL BE INCREMENTED.
11009                                          ;
11010                                  ;*****************************************************************************
  (2)                                 ;TEST 345       TEST THAT IN MODE 2, BAD ADDRESS REFERENCE CAUSES BUS ERROR.
  (3)                                 ;*****************************************************************************
  (2)  024766                         TS345:
11011  024766  012737  025006  000004          MOV     #RETR3,@#RTRAP5 ;SET TRAP RETURN ADDR
11012  024774  012700  177700                  MOV     #177700,R0     ;STORES BAD MEMORY REFERENCE
11013  025000  012720  001234                  MOV     #1234,(R0)+    ;BAD ADDR REFERENCE, TRAP TO LOC 4
11014  025004  104000                          EMT                    ;ADDRESSING 177700 DID NOT CAUSE TRAP
11015  025006  022700  177702          RETR3:  CMP     #177702,R0     ;WAS R0 INCREMENTED?
11016  025012  001401                          BEQ     TS346
  (3)  025014  104000                          EMT                    ;R0 WAS NOT INCREMENTED
11017                                          ;*************************************************************
11018                                          ;
11019                                          ;AFTER THE FIRST BUS ERROR WAS ENCOUNTERED, AN ATTEMPT WAS MADE
11020                                          ;TO PUSH PC AND PS INTO THE STACK. HOWEVER, IF THE STACK POINTER
11021                                          ;WAS BAD, A BOUBLE BUS ERROR OCCURED.  THE STACK POINTER WOULD
11022                                          ;THEN BE SET TO LOCATION 4, OLD PC AND PS WERE PUSHED INTO
11023                                          ;LOCATIONS 0 AND 2. THE PROCESSOR WOULD TRAP TO 4 AND CONTINUE
11024                                          ;EXECUTION.
11025                                          ;
11026                                  ;*****************************************************************************
  (2)                                 ;TEST 346       TEST FOR DOUBLE BUS ERROR.
  (3)                                 ;*****************************************************************************
  (2)  025016                         TS346:
11027  025016  012767  025060  152760          MOV     #DBE1,RTRAP5   ;SET TRAP RETURN ADDR
11028  025024  012737  000340  000006          MOV     #340,@#6       ;SET UP PS
11029  025032  012767  025056  152750          MOV     #DBE2,RTRAP    ;SET TRAP RETURN ADDR
11030  025040  012737  000340  000012          MOV     #340,@#12      ;SET UP PS
```

```
11031  025046  012706  177700            MOV     #177700,SP      ;SET ILLEGAL SP
11032  025052  000077            DBE:    TRAPA                   ;ILLEGAL INSTRUCTION
11033  025054  104000                    EMT                     ;DOUBLE BUS ERROR DID NOT CAUSE TRAP
11034  025056                    DBE2:
  (2)  025056  104000                    EMT                     ;TRAP TO WRONG LOCATION
11035  025060  022737  025054  000000 DBE1:    CMP     #DBE+2,@#0      ;OLD PC GOT SAVED?
11036  025066  001401                    BEQ     DBE3
  (2)  025070  104000                    EMT                     ;OLD PC DID NOT GET SAVEDD
11037  025072  022737  000340  000002 DBE3:    CMP     #340,@#2        ;CORRECT PS SAVED?
11038  025100  001401                    BEQ     DBE4
  (2)  025102  104000                    EMT                     ;CORRECT PS DID NOT GET SAVE
11039  025104  022706  000000            DBE4:   CMP     #0,SP           ;SP POINTS TO LOC 0?
11040  025110  001401                    BEQ     DBE5
  (2)  025112  104000                    EMT                     ;SP IS NOT POINTING TO LOC 0
11041  025114  012706  001000            DBE5:   MOV     #STBOT,SP       ;RESET SP
11042  025120  012767  021526  152656            MOV     #TO4,4          ;RESET VECTOR 4
11043  025126  012767  021530  152654            MOV     #TO10,10        ;RESET VECTOR 10
11044
11045                                      ;*********************************************************************
11046                                      ;THIS TEST WILL CHECK THE SERVICE ROUTINE FOR A CONTROL CHIP ERROR.
11047                                      ;THIS IS DONE BY EXECUTING INSTRUCTIONS WHICH JUMP TO NON-EXISTENT
11048                                      ;CONTROL-CHIP. THE TEST EXECUTES AN FIS INSTRUCTION WHICH
11049                                      ;IS ILLEGAL ON ALL PROCESSORS USING THE DCF11-A CHIP SET.
11050                                      ;A CTLERR TRAPS TO LOCATION 10.
11051                                      ;THE RESET LINE IS ALSO ASSERTED FOR 1 CYCLE.
11052                                  ;*************************************************************************
  (2)                                 ;TEST 347       TEST CTLERR SERVICE ROUTINE
  (3)                                 ;*************************************************************************
  (2)  025134                         TS347:
11053
11054  025134  012706  001000            MOV     #STBOT,R6       ;INIT STACK POINTER
11055  025140  012737  025164  000010    MOV     #1$,@#10                ;SET UP RETURN ADDR FROM TRAP
11056  025146  012737  000340  000012    MOV     #340,@#12       ;SET TRAP PRIORITY=7
11057  025154  075006                    FADD    R6              ;EXECUTE FIS INSTR..SHOULD CAUSE CTLERR
11058  025156  004767  106246            JSR     PC,ABORT        ;SEE IF WE ARE UNDER UFD
11059  025162  000000                    HALT                    ;DID NOT TRAP..CHECK CSEL LINE
11060  025164  012706  001000    1$:     MOV     #STBOT,R6       ;RE-INIT STACK POINTER
11061  025170  012767  021530  152612    MOV     #TO10,10        ;RESET VECTOR 10
11062
11063                                  ;*************************************************************************
  (2)                                 ;TEST 350       TEST THAT ALL RESERVED INSTRUCTIONS TRAP
  (3)                                 ;*************************************************************************
  (2)  025176                         TS350:
11064  025176  042767  000100  152360    BIC     #100,TTCSR
11065                                              ; SET UP TO SEE IF
11066  025204  013767  000010  000042    MOV     @#10,TENSAVE    ;  THIS PROCESSOR HAS THE
11067  025212  012737  025256  000010    MOV     #TRAP10,@#10    ;  FLOATING POINT OPTION
11068  025220  170127  000000            LDFPS   #0              ;DO A FPP INSTRUCTION
11069                                              ;IF NO TRAP FPP INSTALLED
11070  025224  013767  025600  000356    MOV     @#FPP,FINISH    ;SO RESET END OF TABLE POINTER
11071  025232  000411                    BR      AROUND          ; THE FOLLOWING
11072
11073                                  ;* IF NO CIS OPTION TRAP TO HERE
11074  025234  042777  000040  174334 CISTRP: BIC     #40,@#SWR       ;CLEAR CIS OPTION IN SWR
11075  025242  012716  025310            MOV     #CONCIS,(SP)    ;CHANGE RETURN ADDRESS TO CONCIS LOCATION
11076  025246  000002                    RTI                     ;RETURN
```

```
11077  025250  000000                    CISADR: .WORD   0               ;DATA FOR CIS
11078  025252  000000                            .WORD   0               ;               INSTRUCTION
11079  025254  000000                    TENSAVE:.WORD   0               ; A PLACE TO STORE CONTENTS OF 10
11080  025256                            TRAP10:                         ; LEAVE THE TABLE ALONE
11081
11082  025256                            AROUND:                         ;  CONTINUATION POINT
11083  025256  012737  000246  000244            MOV     #246,@#244      ;  RESTORE THE TRAP VECTOR
11084  025264  012737  025234  000010            MOV     #CISTRP,@#10    ;SET UP TO SEE IF THIS HAS THE CIS OPTION
11085  025272  076144                            .WORD   076144          ;EXECUTE A CMPCI INSTRUCTION
11086  025274  025250                            .WORD   CISADR          ;OPERANDS
11087  025276  025250                            .WORD   CISADR          ;         FOR CIS
11088  025300  000000                            .WORD   0               ;                INSTRUCTION
11089  025302  052777  000040  174266            BIS     #40,@SWR        ;SET CIS PRESENT BIT
11090  025310  016737  177740  000010  CONCIS:   MOV     TENSAVE,@#10    ; RESTORE THE ILLEGAL INST. VECTOR
11091  025316  012703  025470                    MOV     #TABLE,TAB      ;TABLE POINTER
11092  025322  012305                    GIN1:   MOV     (TAB)+,FIRST    ;FIRST OR CURRENT INSTRUCTION
11093  025324  012301                            MOV     (TAB)+,LAST     ;LAST INSTRUCTION OR GROUP
11094  025326  020537  025544                    CMP     FIRST,@#CIS
11095  025332  001007                            BNE     1$
11096  025334  032777  000040  174234            BIT     #40,@SWR
11097  025342  001403                            BEQ     1$
11098  025344  012703  025600                    MOV     #FPP,TAB
11099  025350  000764                            BR      GIN1
11100  025352  020567  000232            1$:     CMP     FIRST,FINISH    ;TESTED ALL
11101  025356  001415                            BEQ     GIN3            ;YES BRANCH
11102  025360  010567  000226                    MOV     FIRST,INST      ;SET UP INST
11103  025364  005267  000222            GIN2:   INC     INST
11104  025370  012767  025424  152412            MOV     #RET,10         ;SET UP RETURN FROM TRAP
11105  025376  012706  001000                    MOV     #STBOT,SP       ;SET UP STACK POINTER
11106  025402  005067  152370                    CLR     CC              ;CLEAR PRIORITY
11107  025406  000167  000200                    JMP     INST            ;EXECUTE RESERVED INSTRUCTION
11108  025412  012767  021530  152370  GIN3:   MOV     #T010,10        ;RESET VECTOR 10
11109  025420  000167  000252                    JMP     THRPRT          ;JUMP TO EIS TEST
11110
11111                                    ;TRAPPING SHOULD SEND YOU HERE
11112  025424  020627  000774            RET:    CMP     SP,@#STBOT-4    ;TEST DECREMENT OF SP
11113  025430  001401                            BEQ     RET1
11114  025432  104000                            EMT                     ;WRONG DECREMENT
11115  025434  026727  153334  025614    RET1:   CMP     STBOT-4,@#INST+2 ;LOC OF INST UNINCREMENTED
11116  025442  001401                            BEQ     RET2
11117  025444  104000                            EMT                     ;INST INC ON TRAP
11118  025446  005767  153324            RET2:   TST     STBOT-2
11119  025452  001401                            BEQ     RET3
  (1)  025454                            RET4:
  (2)  025454  104000                            EMT                     ;CONDITION CODES SET ON TRAP OR WRONG $TSTNM
11120  025456  026701  000130            RET3:   CMP     INST,LAST
11121  025462  001717                            BEQ     GIN1            ;SET UP NEW GROUP
11122  025464  000167  177674                    JMP     GIN2            ;FINISH OLD GROUP
11123                                                                    ;END OF INSTRUCTION GROUP
11124  025470  000007                    TABLE:  7                       ;END OF OPERATE
11125  025472  000077                            77
11126  025474  000207                            207                     ;RTS,RT1,JMP
11127  025476  000227                            227
11128  025500  006777                            6777
11129  025502  007777                            7777
11130  025504  075037                            075037
```

```
11131  025506  076017                        76017
11132  025510  076032                        76032
11133  025512  076037                        76037
11134  025514  076045                        76045
11135  025516  076047                        76047
11136  025520  076077                        76077
11137  025522  076127                        76127
11138  025524  076132                        76132
11139  025526  076137                        76137
11140  025530  076145                        76145
11141  025532  076147                        76147
11142  025534  076157                        76157
11143  025536  076167                        76167
11144  025540  076177                        76177
11145  025542  076777                        76777
11146  025544  076017                CIS:    76017
11147  025546  076032                        76032
11148  025550  076037                        76037
11149  025552  076045                        76045
11150  025554  076047                        76047
11151  025556  076077                        76077
11152  025560  076127                        76127
11153  025562  076132                        76132
11154  025564  076137                        76137
11155  025566  076145                        76145
11156  025570  076147                        76147
11157  025572  076157                        76157
11158  025574  076167                        76167
11159  025576  076177                        76177
11160  025600  167777                FPP:    167777                  ;  START OF THE FPP INSTRUCTIONS
11161  025602  177700                        177700
11162  025604  177716                        177716
11163  025606  177777                        177777
11164  025610  025610                FINISH: .                       ;END FLAG
11165  025612  000000                INST:   HALT                    ;WILL CONTINUE RESERVED INST
11166  025614  000000                        HALT                    ;SHOULD TRAP TO LOC 10
11167  025616  000000                        HALT                    ;LOC 10 SHOULD SEND YOU TO
11168  025620  000000                        HALT                    ;RET
11169  025622  000000                        HALT
11170                                .SBTTL ** STARTING OF EIS TEST **
11171
11172
11173          000000                DUMMY=  0
11174          000051                F=      51
11175          000176                N=      176
11176
11177  025624                        COUNT:
11178          025626                        .=COUNT+2
11179  025626                        PSWORD:
11180          025630                        .=PSWORD+2
11181  025630                        TEMP1:
11182          025632                        .=TEMP1+2
11183  025632                        TEMP2:
11184          025634                        .=TEMP2+2
11185  025634                        TEMP3:
11186          025636                        .=TEMP3+2
```

```
11187  025636                          TEMP4:
11188         025640                           .=TEMP4+2
11189  025640 000000                   TEMP5:  .WORD
11190  025642 000000                   TEMP6:  .WORD
11191  025644 177771                   S1:     -7
11192  025646 025644                   S2:     S1
11193  025650 177772                   S3:     -6
11194  025652 177777                   S4:     -1
11195  025654 040000                   S5:     40000
11196  025656 025654                   S6:     S5
11197  025660 040000                   S7:     40000
11198  025662 177776                   S8:     -2
11199  025664 000002                   S9:     2
11200  025666 025664                   S10:    S9
11201  025670 000002                   S11:    2
11202  025672 177566                   $TPB:   177566
11203  025674 177564                   $TPS:   177564
11204
11205
11206
11207
11208
11209
11210
```

```
11321  025676                          THRPRT:
11322
11323  025676  012705  001004                  MOV     #$TESTN,R5      ;MAKE R5 POINT TO WHERE TEST # IS SAVED
11324  025702  005037  025624                  CLR     @#COUNT         ;CLEAR THE COUNTER
11325  025706  012715  000001                  MOV     #1,(R5)         ;INITIALIZE TEST NUMBER
11326  025712  012706  001000                  MOV     #STBOT,SP       ;** STACK AT STBOT **
11327  025716  012737  000001  025630  2$:     MOV     #1,@#TEMP1      ;TEMP1=1
11328  025724  005037  025632                  CLR     @#TEMP2         ;TEMP2=0
11329  025730  012737  000001  025634          MOV     #1,@#TEMP3      ;TEMP3=1
11330  025736  005037  025636                  CLR     @#TEMP4         ;TEMP4=0
11331  025742  106427  000000                  MTPS    #0
11332
11333
```

```
11338
11339                                   ;**********************************
11340                                   ;     ASH INSTRUCTION TESTS
11341                                   ;**********************************
11342
11343
11344
11345
11346                                           ;*************
11347                                           ; TESTS 1-36
11348                                           ;*************
11349
11350
11351  025746  013700  025630        ASTART: MOV     @#TEMP1,#0      ;LOAD R0 WITH THE CONTENTS OF TEMP1
11352  025752  032737  000001  001006        BIT     #1,@#$PASS      ;IS IT AN EVEN PASS ?
11353  025760  001004                        BNE     2$              ;IF NOT THEN GO TO 2$
11354  025762  013701  025632                MOV     @#TEMP2,R1      ;OTHERWISE EXECUTE THE INSTRUCTION
11355                                                                 ;IN MODE 0 USING R1
11356  025766  072001                        ASH     R1,R0
11357  025770  000402                        BR      4$
11358  025772  072067  177634        2$:     ASH     TEMP2,#0        ;SHIFT R0 BY THE NUMBER SPECIFIED BY TEMP2
11362  025776  106737  025626        4$:     MFPS    @#PSWORD        ;SAVE PS
11366  026002  123737  025636  025626        CMPB    @#TEMP4,@#PSWORD;IS THE PS = TEMP4 ?
11367  026010  001401                        BEQ     11$
  (2)  026012  104000                        EMT                     ;THE PS IS NOT EQUAL TO 0
11368  026014  005237  025624        11$:    INC     @#COUNT         ;INCREMENT THE COUNTER
11369  026020  023700  025634                CMP     @#TEMP3,#0      ;IS THE RESULT IN R0 EQUAL TO TEMP3?
11370  026024  001401                        BEQ     12$
  (1)  026026                        6$:
  (2)  026026  104000                        EMT                     ;EITHER INCORRECT R0 OR INCORRECT SEQUENCE
11371  026030  021537  025624        12$:    CMP     (R5),@#COUNT    ;IS THE TEST NUMBER EQUAL TO THE
11372                                                                 ;COUNTER?
11373  026034  001374                        BNE     6$              ;IF NOT GO TO THE HLT ABOVE
11374  026036  005215                        INC     (R5)
11375  026040  021527  000037                CMP     (R5),#37        ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT
11376                                                                 ;BY 14. AND RIGHT BY 14.?
11377  026044  002011                        BGE     8$
11378  026046  005237  025632                INC     @#TEMP2
11379  026052  006367  177556                ASL     TEMP3           ;SHIFT TEMP3 LEFT.
11380  026056  021527  000020                CMP     (R5),#20        ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
11381  026062  001004                        BNE     REGR1
11382  026064  000167  000670                JMP     NEGAT           ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
11383  026070  004767  000712        8$:     JSR     PC,TST37        ;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
11384  026074  013701  025630        REGR1:  MOV     @#TEMP1,#1      ;LOAD R1 WITH THE CONTENTS OF TEMP1
11385  026100  032737  000001  001006        BIT     #1,@#$PASS      ;IS IT AN EVEN PASS ?
11386  026106  001004                        BNE     2$              ;IF NOT THEN GO TO 2$
11387  026110  013702  025632                MOV     @#TEMP2,R2      ;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
11388  026114  072102                        ASH     R2,R1           ;USING R1
11389  026116  000402                        BR      4$
11390  026120  072167  177506        2$:     ASH     TEMP2,#1        ;SHIFT R1 BY THE NUMBER SPECIFIED BY TEMP2
11394  026124  106737  025626        4$:     MFPS    @#PSWORD        ;SAVE PS
11398  026130  123737  025636  025626        CMPB    @#TEMP4,@#PSWORD;IS THE PS = TEMP4 ?
11399  026136  001401                        BEQ     11$
  (2)  026140  104000                        EMT                     ;THE PS IS NOT EQUAL TO 0
11400  026142  005237  025624        11$:    INC     @#COUNT         ;INCREMENT THE COUNTER
11401  026146  023701  025634                CMP     @#TEMP3,#1      ;IS THE RESULT IN R1 EQUAL TO TEMP3?
```

```
11402  026152  001401                        BEQ     12$
  (1)  026154                          6$:
  (2)  026154  104000                        EMT                     ;EITHER INCORRECT R1 OR INCORRECT SEQUENCE
11403  026156  021537  025624          12$:  CMP     (R5),@#COUNT      ;IS THE TEST NUMBER EQUAL TO THE COUNTER?
11404  026162  001374                        BNE     6$                ;IF NOT GO TO THE HLT ABOVE
11405  026164  005215                        INC     (R5)
11406  026166  021527  000037                CMP     (R5),#37          ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT
11407                                                                  ;BY 14. AND RIGHT BY 14.?
11408  026172  002011                        BGE     8$
11409  026174  005237  025632                INC     @#TEMP2
11410  026200  006367  177430                ASL     TEMP3             ;SHIFT TEMP3 LEFT
11411  026204  021527  000020                CMP     (R5),#20          ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
11412  026210  001004                        BNE     REGR2
11413  026212  000167  000542                JMP     NEGAT             ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
11414  026216  004767  000564          8$:   JSR     PC,TST37          ;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
11415  026222  013702  025630          REGR2: MOV    @#TEMP1,R2        ;LOAD R2 WITH THE CONTENTS OF TEMP1
11416  026226  032737  000001  001006        BIT     #1,@#$PASS        ;IS IT AN EVEN PASS ?
11417  026234  001004                        BNE     2$                ;IF NOT THEN GO TO 2$
11418  026236  013703  025632                MOV     @#TEMP2,R3        ;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
11419  026242  072203                        ASH     R3,R2             ;USING R2
11420  026244  000402                        BR      4$
11421  026246  072267  177360          2$:   ASH     TEMP2,R2          ;SHIFT R2 BY THE NUMBER SPECIFIED BY TEMP2
11425  026252  106737  025626          4$:   MFPS    @#PSWORD          ;SAVE PS
11429  026256  123737  025636  025626        CMPB    @#TEMP4,@#PSWORD;IS THE PS = TEMP4 ?
11430  026264  001401                        BEQ     11$
  (2)  026266  104000                        EMT                     ;THE PS IS NOT EQUAL TO 0
11431  026270  005237  025624          11$:  INC     @#COUNT
11432  026274  023702  025634                CMP     @#TEMP3,R2        ;IS THE RESULT IN R2 EQUAL TO TEMP3?
11433  026300  001401                        BEQ     12$
  (1)  026302                          6$:
  (2)  026302  104000                        EMT                     ;EITHER INCORRECT R2 OR INCORRECT SEQUENCE
11434  026304  021537  025624          12$:  CMP     (R5),@#COUNT      ;IS THE TEST NUMBER EQUAL TO THE COUNTER?
11435  026310  001374                        BNE     6$                ;IF NOT GO TO THE HLT ABOVE
11436  026312  005215                        INC     (R5)
11437  026314  021527  000037                CMP     (R5),#37          ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED
11438                                                                  ;LEFT BY 14. AND RIGHT BY 14.?
11439  026320  002011                        BGE     8$
11440  026322  005237  025632                INC     @#TEMP2
11441  026326  006367  177302                ASL     TEMP3             ;SHIFTED TEMP3 LEFT
11442  026332  021527  000020                CMP     (R5),#20          ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
11443  026336  001004                        BNE     REGR3
11444  026340  000167  000414                JMP     NEGAT             ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
11445  026344  004767  000436          8$:   JSR     PC,TST37          ;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
11446  026350  013703  025630          REGR3: MOV    @#TEMP1,R3        ;LOAD R3 WITH THE CONTENTS OF TEMP1
11447  026354  032737  000001  001006        BIT     #1,@#$PASS        ;IS IT AN EVEN PASS ?
11448  026362  001004                        BNE     2$                ;IF NOT THEN GO TO 2$
11449  026364  013704  025632                MOV     @#TEMP2,R4        ;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
11450  026370  072304                        ASH     R4,R3             ;USING R3
11451  026372  000402                        BR      4$
11452  026374  072367  177232          2$:   ASH     TEMP2,R3          ;SHIFT R3 BY THE NUMBER SPECIFIED BY TEMP2
11456  026400  106737  025626          4$:   MFPS    @#PSWORD          ;SAVE PS
11460  026404  123737  025636  025626        CMPB    @#TEMP4,@#PSWORD;IS THE PS = TEMP4 ?
11461  026412  001401                        BEQ     11$
  (2)  026414  104000                        EMT                     ;THE PS IS NOT EQUAL TO 0.
11462  026416  005237  025624          11$:  INC     @#COUNT
11463  026422  023703  025634                CMP     @#TEMP3,R3        ;IS THE RESULT IN R3 EQUAL TO TEMP3?
```

```
11464  026426  001401                          BEQ     12$
  (1)  026430                          6$:
  (2)  026430  104000                          EMT                             ;EITHER INCORRECT R3 OR INCORRECT SEQUENCE
11465  026432  021537  025624          12$:    CMP     (R5),@#COUNT            ;IS THE TEST NUMBER EQUAL TO THE COUNTER?
11466  026436  001374                          BNE     6$                      ;IF NOT GO TO THE HLT ABOVE
11467  026440  005215                          INC     (R5)
11468  026442  021527  000037                  CMP     (R5),#37                ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED
11469                                                                          ;LEFT BY 14. AND RIGHT BY 14.?
11470  026446  002010                          BGE     8$
11471  026450  005237  025632                  INC     @#TEMP2
11472  026454  006367  177154                  ASL     TEMP3                   ;SHIFT TEMP3 LEFT?
11473  026460  021527  000020                  CMP     (R5),#20                ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
11474  026464  001003                          BNE     REGR4
11475  026466  000534                          BR      NEGAT                   ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
11476  026470  004767  000312          8$:     JSR     PC,TST37                ;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
11477  026474  013704  025630          REGR4:  MOV     @#TEMP1,%4              ;LOAD R4 WITH THE CONTENTS OF TEMP1
11478  026500  010501                          MOV     R5,R1                   ;SAVE R5
11479  026502  032737  000001  001006          BIT     #1,@#$PASS              ;IS IT AN EVEN PASS ?
11480  026510  001004                          BNE     2$                      ;IF NOT THEN GO TO 2$
11481  026512  013705  025632                  MOV     @#TEMP2,R5              ;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
11482  026516  072405                          ASH     R5,R4                   ;USING R4
11483  026520  000402                          BR      4$
11484  026522  072467  177104          2$:     ASH     TEMP2,%4                ;SHIFT R4 BY THE NUMBER SPECIFIED BY TEMP2
11488  026526  106737  025626          4$:     MFPS    @#PSWORD                ;SAVE PS
11492  026532  123737  025636  025626          CMPB    @#TEMP4,@#PSWORD;IS PS = TEMP4 ?
11493  026540  001401                          BEQ     11$
  (2)  026542  104000                          EMT                             ;THE PS IS NOT EQUAL TO 0
11494  026544  005237  025624          11$:    INC     @#COUNT
11495  026550  023704  025634                  CMP     @#TEMP3,%4              ;IS THE RESULT IN R4 EQUAL TO TEMP3?
11496  026554  001401                          BEQ     12$
  (1)  026556                          6$:
  (2)  026556  104000                          EMT                             ;EITHER INCORRECT R4 OR INCORRECT SEQUENCE
11497  026560  010105                  12$:    MOV     R1,R5                   ;RESTORE R5
11498  026562  021537  025624                  CMP     (R5),@#COUNT            ;IS THE TEST NUMBER EQUAL TO THE COUNTER?
11499  026566  001373                          BNE     6$                      ;IF NOT GO TO THE HLT ABOVE
11500  026570  005215                          INC     (R5)
11501  026572  021527  000037                  CMP     (R5),#37                ;HAS THE CONTENTS OF REGISTERS BEEN
11502                                                                          ;SHIFTED LEFT BY 14. AND RIGHT BY 14.?
11503  026576  002010                          BGE     8$
11504  026600  005237  025632                  INC     @#TEMP2
11505  026604  006367  177024                  ASL     TEMP3                   ;SHIFT TEMP3 LEFT
11506  026610  021527  000020                  CMP     (R5),#20                ;HAS THE CONTENTS OF REGISTER BEEN SHIFTED BY 14.?
11507  026614  001003                          BNE     REGR5
11508  026616  000460                          BR      NEGAT                   ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
11509  026620  004767  000162          8$:     JSR     PC,TST37                ;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
11510  026624  010501                  REGR5:  MOV     R5,R1                   ;SAVE R5
11511  026626  013705  025630                  MOV     @#TEMP1,%5              ;LOAD R5 WITH THE CONTENTS OF TEMP1
11512  026632  032737  000001  001006          BIT     #1,@#$PASS              ;IS IT AN EVEN PASS ?
11513  026640  001004                          BNE     2$                      ;IF NOT THEN GO TO 2$
11514  026642  013700  025632                  MOV     @#TEMP2,R0              ;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
11515  026646  072500                          ASH     R0,R5                   ;USING R5
11516  026650  000402                          BR      4$
11517  026652  072567  176754          2$:     ASH     TEMP2,%5                ;SHIFT R5 BY THE NUMBER SPECIFIED BY TEMP2
11521  026656  106737  025626          4$:     MFPS    @#PSWORD                ;SAVE PS
11525  026662  123737  025636  025626          CMPB    @#TEMP4,@#PSWORD;IS PS = TEMP4 ?
11526  026670  001401                          BEQ     11$
```

```
  (2)  026672  104000                          EMT                          ;THE PS IS NOT EQUAL TO 0.
11527  026674  005237  025624          11$:    INC     @#COUNT
11528  026700  023705  025634                  CMP     @#TEMP3,#5           ;IS THE RESULT IN R5 EQUAL TO TEMP3?
11529  026704  001401                          BEQ     12$
  (1)  026706                          6$:
  (2)  026706  104000                          EMT                          ;EITHER INCORRECT R5 OR INCORRECT SEQUENCE
11530  026710  021137  025624          12$:    CMP     (R1),@#COUNT         ;IS THE TEST NUMBER EQUAL TO THE COUNTER?
11531  026714  001374                          BNE     6$                   ;IF NOT GO TO THE HLT ABOVE
11532  026716  010105                          MOV     R1,R5                ;RESTORE R5
11533  026720  005215                          INC     (R5)
11534  026722  021527  000037                  CMP     (R5),#37             ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED
11535                                                                       ;LEFT BY 14. AND RIGHT BY 14.?
11536  026726  002010                          BGE     8$                   ;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
11537  026730  005237  025632                  INC     @#TEMP2
11538  026734  006367  176674                  ASL     TEMP3                ;SHIFT TEMP3 LEFT
11539  026740  021527  000020                  CMP     (R5),#20             ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
11540  026744  001405                          BEQ     NEGAT                ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
11541  026746  000402                          BR      10$
11542  026750  004767  000032          8$:     JSR     PC,TST37
11543  026754  000167  176766          10$:    JMP     ASTART               ;GO BACK TO START
11544  026760  012737  040000  025630  NEGAT:  MOV     #40000,@#TEMP1       ;TEMP1=40000
11545  026766  012737  177762  025632          MOV     #177762,@#TEMP2      ;TEMP2=177762
11546  026774  012737  000001  025634          MOV     #1,@#TEMP3           ;TEMP3=1
11547  027002  000167  176740                  JMP     ASTART
11548  027006  021527  000037          TST37:  CMP     (R5),#37             ;IS IT TEST 37?
11549  027012  001013                          BNE     TST40                ;IF NOT THEN TRY TEST 40
11550  027014  005037  025630                  CLR     @#TEMP1              ;0
11551  027020  012737  000020  025632          MOV     #16.,@#TEMP2         ;SHIFTED BY 16
11552  027026  005037  025634                  CLR     @#TEMP3              ;IS=0
11553  027032  012737  000004  025636          MOV     #4,@#TEMP4           ;AND PS=4
11554  027040  000207                          RTS     PC
11555  027042  021527  000040          TST40:  CMP     (R5),#40             ;IS IT TEST 40?
11556  027046  001003                          BNE     TST41                ;IF NOT THEN TRY TEST 41
11557  027050  005037  025632                  CLR     @#TEMP2              ;0 SHIFTED BY 0=0 AND PS=4
11558  027054  000207                          RTS     PC
11559  027056  021527  000041          TST41:  CMP     (R5),#41             ;IS IT TEST 41?
11560  027062  001004                          BNE     TST42                ;IF NOT THEN TRY TEST 42
11561  027064  012737  177760  025632          MOV     #-16.,@#TEMP2        ;0 SHIFTED BY -16.=0 AND PS=4
11562  027072  000207                          RTS     PC
11563  027074  021527  000042          TST42:  CMP     (R5),#42             ;IS IT TEST 42?
11564  027100  001013                          BNE     TST43                ;IF NOT THEN TRY TEST 43
11565  027102  012737  100000  025630          MOV     #100000,@#TEMP1      ;100000
11566  027110  005237  025632                  INC     @#TEMP2              ;SHIFTED BY -15
11567  027114  005337  025634                  DEC     @#TEMP3              ;IS=-1
11568  027120  012737  000010  025636          MOV     #10,@#TEMP4                  ;AND PS=10
11569  027126  000207                          RTS     PC
11570  027130  021527  000043          TST43:  CMP     (R5),#43             ;IS IT TEST 43?
11571  027134  001012                          BNE     TST44                ;IF NOT THEN IF NOT THEN TRY TEST 44
11572  027136  012737  125252  025630          MOV     #125252,@#TEMP1      ;125252
11573  027144  012737  177777  025632          MOV     #-1,@#TEMP2          ;SHIFTED BY -1
11574  027152  012737  152525  025634          MOV     #152525,@#TEMP3      ;IS=152525 AND PS=10
11575  027160  000207                          RTS     PC
11576  027162  021527  000044          TST44:  CMP     (R5),#44             ;IS IT TEST 44?
11577  027166  001012                          BNE     TST45                ;IF NOT THEN TRY TEST 45
11578  027170  012737  000001  025632          MOV     #1,@#TEMP2           ;125252 SHIFTED BY 1
11579  027176  012737  052524  025634          MOV     #52524,@#TEMP3       ;IS=52524
```

```
11580  027204  012737  000003  025636          MOV     #3,@#TEMP4      ;AND PS=3
11581  027212  000207                          RTS     PC
11582  027214  021527  000045          TST45:  CMP     (R5),#45        ;IS IT TEST 45?
11583  027220  001012                          BNE     TST46           ;IF NOT THEN TRY TEST 46
11584  027222  012737  177776  025632          MOV     #-2,@#TEMP2     ;125252 SHIFTED BY -2
11585  027230  012737  165252  025634          MOV     #165252,@#TEMP3 ;IS=165252
11586  027236  012737  000011  025636          MOV     #11,@#TEMP4     ;AND PS=11
11587  027244  000207                          RTS     PC
11588  027246  021527  000046          TST46:  CMP     (R5),#46        ;IS IT TEST 46?
11589  027252  001014                          BNE     TST47           ;IF NOT THEN TRY TEST 47
11590  027254  012737  177777  025630          MOV     #-1,@#TEMP1     ;-1
11591  027262  012737  000020  025632          MOV     #16.,@#TEMP2    ;SHIFTED BY 15.
11592  027270  005037  025634                  CLR     @#TEMP3         ;IS=0
11593  027274  012737  000007  025636          MOV     #7,@#TEMP4      ;AND PS=7
11594  027302  000207                          RTS     PC
11595  027304  021527  000047          TST47:  CMP     (R5),#47        ;IS IT TEST 47?
11596  027310  001011                          BNE     TST50           ;IF NOT THEN TRY TEST 50
11597  027312  005337  025632                  DEC     @#TEMP2         ;-1 SHIFTED BY 15
11598  027316  012737  100000  025634          MOV     #100000,@#TEMP3 ;IS=100000
11599  027324  012737  000011  025636          MOV     #11,@#TEMP4     ;AND PS=11
11600  027332  000207                          RTS     PC
11601  027334  021527  000050          TST50:  CMP     (R5),#50        ;IS IT TEST 50
11602  027340  001007                          BNE     ENT51           ;IF NOT THEN TRY TEST 51
11603  027342  012737  137777  025630          MOV     #137777,@#TEMP1 ;137777 SHIFTED BY 15. IS=100000
11604  027350  012737  000013  025636          MOV     #13,@#TEMP4     ;AND PS=13
11605  027356  000207                          RTS     PC
11606  027360  021527  000051          ENT51:  CMP     (R5),#51        ;IS IT ENTERING TEST 51?
11607  027364  001401                          BEQ     1$
  (2)  027366  104000                          EMT                     ;TEST NUMBER GOOFED
11608
11609  027370  005726                  1$:     TST     (SP)+           ;RESTORE STACK POINTER
11610  027372  012704  177771                  MOV     #-7,%4
11611  027376  012702  025644                  MOV     #S1,%2
11612  027402  012703  025646                  MOV     #S2,%3
11613                                  ;******************************************************************************
  (1)                               ;TEST:51   11/34   ASH     125252 SHIFTED BY #5 = 52500  PS = 3
  (1)                               ;******************************************************************************
  (1)
  (1)  027406  012701  125252          TST51:  MOV     #125252,%1      ;LOAD R1 WITH 125252
  (1)  027412  072127  000005                  ASH     #5,%1           ;SHIFT R1 BY #5
  (1)  027416  106737  025626                  MFPS    @#PSWORD        ;SAVE PS
  (1)  027422  122737  000003  025626          CMPB    #3,@#PSWORD     ;IS THE PS 3?
  (2)  027430  001401                          BEQ     11$
  (3)  027432  104000                          EMT                     ;THE PS IS NOT EQUAL TO 3
  (1)  027434  022701  052500          11$:    CMP     #52500,%1       ;IS THE RESULT 52500?
  (2)  027440  001401                          BEQ     12$
  (3)  027442  104000                          EMT                     ;R1 IS NOT EQUAL TO 52500 OR INCORRECT SEQUENCE
  (1)  027444  005215                  12$:    INC     (R5)
  (1)
  (1)
  (1)
11614                                   ;******************************************************************************
  (1)                               ;TEST:52   11/34   ASH     125252 SHIFTED BY @S2 = 177525  PS = 10
  (1)                               ;******************************************************************************
  (1)
  (1)  027446  012700  125252          TST52:  MOV     #125252,%0      ;LOAD R0 WITH 125252
```

```
  (1)  027452  072077  176170                ASH     @S2,#0          ;SHIFT R0 BY @S2
  (1)  027456  106737  025626                MFPS    @@PSWORD        ;SAVE PS
  (1)  027462  122737  000010  025626        CMPB    #10,@@PSWORD    ;IS THE PS 10?
  (2)  027470  001401                        BEQ     11$
  (3)  027472  104000                        EMT                     ;THE PS IS NOT EQUAL TO 10
  (1)  027474  022700  177525          11$:  CMP     #177525,#0      ;IS THE RESULT 177525?
  (2)  027500  001401                        BEQ     12$
  (3)  027502  104000                        EMT                     ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
  (1)  027504  005215                12$:    INC     (R5)
  (1)
  (1)
  (1)
11615                                ;*************************************************************************
  (1)                                ;TEST:53   11/34   ASH     125252 SHIFTED BY @@S1 = 177525  PS = 10
  (1)                                ;*************************************************************************
  (1)
  (1)  027506  012700  125252          TST53:  MOV     #125252,#0      ;LOAD R0 WITH 125252
  (1)  027512  072037  025644                  ASH     @@S1,#0         ;SHIFT R0 BY @@S1
  (1)  027516  106737  025626                  MFPS    @@PSWORD        ;SAVE PS
  (1)  027522  122737  000010  025626          CMPB    #10,@@PSWORD    ;IS THE PS 10?
  (2)  027530  001401                          BEQ     11$
  (3)  027532  104000                          EMT                     ;THE PS IS NOT EQUAL TO 10
  (1)  027534  022700  177525            11$:  CMP     #177525,#0      ;IS THE RESULT 177525?
  (2)  027540  001401                          BEQ     12$
  (3)  027542  104000                          EMT                     ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
  (1)  027544  005215                  12$:    INC     (R5)
  (1)
  (1)
  (1)
11616                                ;*************************************************************************
  (1)                                ;TEST:54   11/34   ASH     125252 SHIFTED BY (2) = 177525  PS = 10
  (1)                                ;*************************************************************************
  (1)
  (1)  027546  012700  125252          TST54:  MOV     #125252,#0      ;LOAD R0 WITH 125252
  (1)  027552  072012                          ASH     (2),#0          ;SHIFT R0 BY (2)
  (1)  027554  106737  025626                  MFPS    @@PSWORD        ;SAVE PS
  (1)  027560  122737  000010  025626          CMPB    #10,@@PSWORD    ;IS THE PS 10?
  (2)  027566  001401                          BEQ     11$
  (3)  027570  104000                          EMT                     ;THE PS IS NOT EQUAL TO 10
  (1)  027572  022700  177525            11$:  CMP     #177525,#0      ;IS THE RESULT 177525?
  (2)  027576  001401                          BEQ     12$
  (3)  027600  104000                          EMT                     ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
  (1)  027602  005215                  12$:    INC     (R5)
  (1)
  (1)
  (1)
11617                                ;*************************************************************************
  (1)                                ;TEST:55   11/34   ASH     125252 SHIFTED BY (2)+ = 177525  PS = 10
  (1)                                ;*************************************************************************
  (1)
  (1)  027604  012700  125252          TST55:  MOV     #125252,#0      ;LOAD R0 WITH 125252
  (1)  027610  072022                          ASH     (2)+,#0         ;SHIFT R0 BY (2)+
  (1)  027612  106737  025626                  MFPS    @@PSWORD        ;SAVE PS
  (1)  027616  122737  000010  025626          CMPB    #10,@@PSWORD    ;IS THE PS 10?
  (2)  027624  001401                          BEQ     11$
  (3)  027626  104000                          EMT                     ;THE PS IS NOT EQUAL TO 10
```

```
  (1)  027630  022700  177525          11$:    CMP     #177525,#0      ;IS THE RESULT 177525?
  (2)  027634  001401                          BEQ     12$
  (3)  027636  104000                          EMT                      ;RO IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
  (1)  027640  005215                  12$:    INC     (R5)
  (1)
  (1)
  (1)
11618                                   ;*************************************************************************
  (1)                                   ;TEST:56   11/34   ASH     125252 SHIFTED BY -(2) = 177525  PS = 10
  (1)                                   ;*************************************************************************
  (1)
  (1)  027642  012700  125252          TST56:  MOV     #125252,#0       ;LOAD RO WITH 125252
  (1)  027646  072042                          ASH     -(2),#0          ;SHIFT RO BY -(2)
  (1)  027650  106737  025626                  MFPS    @#PSWORD         ;SAVE PS
  (1)  027654  122737  000010  025626          CMPB    #10,@#PSWORD     ;IS THE PS 10?
  (2)  027662  001401                          BEQ     11$
  (3)  027664  104000                          EMT                      ;THE PS IS NOT EQUAL TO 10
  (1)  027666  022700  177525          11$:    CMP     #177525,#0       ;IS THE RESULT 177525?
  (2)  027672  001401                          BEQ     12$
  (3)  027674  104000                          EMT                      ;RO IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
  (1)  027676  005215                  12$:    INC     (R5)
  (1)
  (1)
  (1)
11619                                   ;*************************************************************************
  (1)                                   ;TEST:57   11/34   ASH     125252 SHIFTED BY 2(3) = 177252  PS = 11
  (1)                                   ;*************************************************************************
  (1)
  (1)  027700  012700  125252          TST57:  MOV     #125252,#0       ;LOAD RO WITH 125252
  (1)  027704  072063  000002                  ASH     2(3),#0          ;SHIFT RO BY 2(3)
  (1)  027710  106737  025626                  MFPS    @#PSWORD         ;SAVE PS
  (1)  027714  122737  000011  025626          CMPB    #11,@#PSWORD     ;IS THE PS 11?
  (2)  027722  001401                          BEQ     11$
  (3)  027724  104000                          EMT                      ;THE PS IS NOT EQUAL TO 11
  (1)  027726  022700  177252          11$:    CMP     #177252,#0       ;IS THE RESULT 177252?
  (2)  027732  001401                          BEQ     12$
  (3)  027734  104000                          EMT                      ;RO IS NOT EQUAL TO 177252 OR INCORRECT SEQUENCE
  (1)  027736  005215                  12$:    INC     (R5)
  (1)
  (1)
  (1)
11620                                   ;*************************************************************************
  (1)                                   ;TEST:60   11/34   ASH     125252 SHIFTED BY 0(3) = 177525  PS = 10
  (1)                                   ;*************************************************************************
  (1)
  (1)  027740  012700  125252          TST60:  MOV     #125252,#0       ;LOAD RO WITH 125252
  (1)  027744  072073  000000                  ASH     0(3),#0          ;SHIFT RO BY 0(3)
  (1)  027750  106737  025626                  MFPS    @#PSWORD         ;SAVE PS
  (1)  027754  122737  000010  025626          CMPB    #10,@#PSWORD     ;IS THE PS 10?
  (2)  027762  001401                          BEQ     11$
  (3)  027764  104000                          EMT                      ;THE PS IS NOT EQUAL TO 10
  (1)  027766  022700  177525          11$:    CMP     #177525,#0       ;IS THE RESULT 177525?
  (2)  027772  001401                          BEQ     12$
  (3)  027774  104000                          EMT                      ;RO IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
  (1)  027776  005215                  12$:    INC     (R5)
  (1)
```

```
 (1)
 (1)
11621                                   ;*****************************************************************
 (1)                                    ;TEST:61   11/34   ASH     125252 SHIFTED BY @(3)+ = 177525  PS = 10
 (1)                                    ;*****************************************************************
 (1)
 (1)  030000  012700  125252            TST61:  MOV     #125252,%0      ;LOAD R0 WITH 125252
 (1)  030004  072033                            ASH     @(3)+,%0        ;SHIFT R0 BY @(3)+
 (1)  030006  106737  025626                    MFPS    @#PSWORD        ;SAVE PS
 (1)  030012  122737  000010  025626            CMPB    #10,@#PSWORD    ;IS THE PS 10?
 (2)  030020  001401                            BEQ     11$
 (3)  030022  104000                            EMT                     ;THE PS IS NOT EQUAL TO 10
 (1)  030024  022700  177525            11$:    CMP     #177525,%0      ;IS THE RESULT 177525?
 (2)  030030  001401                            BEQ     12$
 (3)  030032  104000                            EMT                     ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
 (1)  030034  005215                    12$:    INC     (R5)
 (1)
 (1)
 (1)
11622                                   ;*****************************************************************
 (1)                                    ;TEST:62   11/34   ASH     125252 SHIFTED BY @-(3) = 177525  PS = 10
 (1)                                    ;*****************************************************************
 (1)
 (1)  030036  012700  125252            TST62:  MOV     #125252,%0      ;LOAD R0 WITH 125252
 (1)  030042  072053                            ASH     @-(3),%0        ;SHIFT R0 BY @-(3)
 (1)  030044  106737  025626                    MFPS    @#PSWORD        ;SAVE PS
 (1)  030050  122737  000010  025626            CMPB    #10,@#PSWORD    ;IS THE PS 10?
 (2)  030056  001401                            BEQ     11$
 (3)  030060  104000                            EMT                     ;THE PS IS NOT EQUAL TO 10
 (1)  030062  022700  177525            11$:    CMP     #177525,%0      ;IS THE RESULT 177525?
 (2)  030066  001401                            BEQ     12$
 (3)  030070  104000                            EMT                     ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
 (1)  030072  005215                    12$:    INC     (R5)
 (1)
 (1)
 (1)
```

```
1627                                    ;**********************************
1628                                    ;     ASHC INSTRUCTION TESTS
1629                                    ;**********************************
1630
1631
1632
1633                                            ;************
1634                                            ;TESTS 63-157
1635                                            ;************
1636
1637
1638
1639  030074  012737  000062  025624          MOV     #62,@#COUNT
1640  030102  005037  025630                  CLR     @#TEMP1         ;TEMP1=0
1641  030106  012737  000001  025632          MOV     #1,@#TEMP2      ;TEMP2=1
1642  030114  005037  025634                  CLR     @#TEMP3         ;TEMP3=0
1643  030120  005037  025636                  CLR     @#TEMP4         ;TEMP4=0
1644  030124  012737  000001  025640          MOV     #1,@#TEMP5      ;TEMP5=1
1645  030132  005037  025642                  CLR     @#TEMP6         ;0 1 SHIFTED BY 0=0 1, PS=0
1646
1647  030136  010502                  REG01:  MOV     R5,R2           ;SAVE R5
1648  030140  013700  025630                  MOV     @#TEMP1,#0      ;PLACE THE CONTENTS OF TEMP1 IN REGISTER 0
1649  030144  013701  025632                  MOV     @#TEMP2,#0!1    ;PLACE THE CONTENTS OF TEMP2 IN REGISTER 1
1650  030150  000241                          CLC
1651  030152  032737  000001  001006          BIT     #1,@#$PASS      ;IS IT AN EVEN PASS ?
1652  030160  001004                          BNE     2$              ;IF NOT THEN GO TO 2$
1653  030162  013705  025634                  MOV     @#TEMP3,R5      ;OTHERWISE EXECUTE ASHC INSTRUCTION IN MODE 0
1654  030166  073005                          ASHC    R5,R0           ;USING R0
1655  030170  000402                          BR      4$
1656  030172  073067  175436          2$:     ASHC    TEMP3,#0        ;ASHC REGISTER 0 BY THE CONTENTS OF TEMP3
1660  030176  106737  025626          4$:     MFPS    @#PSWORD        ;SAVE PS
1664  030202  123737  025642  025626          CMPB    @#TEMP6,@#PSWORD;COMPARE PS WITH THE CONTENTS OF TEMP6
1665  030210  001401                          BEQ     11$
 (2)  030212  104000                          EMT                     ;WRONG PS
1666  030214  005237  025624          11$:    INC     @#COUNT
1667  030220  023700  025636                  CMP     @#TEMP4,#0      ;IS THE RESULT IN R0 SAME AS TEMP4?
1668  030224  001401                          BEQ     12$
 (2)  030226  104000                          EMT                     ;WRONG RESULT IN R0
1669  030230  023701  025640          12$:    CMP     @#TEMP5,#1      ;IS THE RESULT IN R1 SAME AS TEMP5?
1670                                          ;TEMP1 TEMP2 SHIFTED BY TEMP3=TEMP4 TEMP5
1671                                          ;AND PS=TEMP6
1672  030234  001401                          BEQ     13$
 (2)  030236  104000                          EMT                     ;WRONG RESULT IN R1
1673  030240  010205          13$:    MOV     R2,R5           ;RESTORE R5
1674  030242  021537  025624                  CMP     (R5),@#COUNT    ;IS TEST NUMBER=COUNTER?
1675  030246  001401                          BEQ     14$
 (2)  030250  104000                          EMT                     ;NO
1676  030252  005215          14$:    INC     (R5)
1677  030254  021527  000160                  CMP     (R5),#160       ;HAVE THE FIRST 159 TEST BEEN EXECUTED?
1678  030260  002014                          BGE     6$              ;YES
1679  030262  005237  025634                  INC     @#TEMP3
1680  030266  000241                          CLC
1681  030270  006137  025640                  ROL     @#TEMP5         ;ROTATE TEMP5 LEFT BY 1 PLACE
1682  030274  006137  025636                  ROL     @#TEMP4         ;INTRODUCE CARRY FROM TEMP4 IN TEMP5
1683  030300  021527  000121                  CMP     (R5),#121       ;IS IT TEST 121?
1684  030304  001004                          BNE     REGR23
```

```
11685  030306  004467  000344              JSR     R4,RITSH        ;IF SO THEN GO AND INITIATE RIGHT SHIFT
11686  030312  004767  000374      6$:     JSR     #7,TST160
11687  030316  013702  025630      REGR23: MOV     @#TEMP1,#2      ;PLACE THE CONTENTS OF TEMP1 IN REGISTER 2
11688  030322  013703  025632              MOV     @#TEMP2,#2!1    ;PLACE THE CONTENTS OF TEMP2 IN REGISTER 3
11689  030326  000241                      CLC
11690  030330  032737  000001  001006      BIT     #1,@#$PASS      ;IS IT AN EVEN PASS ?
11691  030336  001004                      BNE     2$              ;IF NOT THEN GO TO 2$
11692  030340  013704  025634              MOV     @#TEMP3,R4      ;OTHERWISE EXECUTE ASHC INSTRUCTION IN MODE 0
11693  030344  073204                      ASHC    R4,R2           ;USING R2
11694  030346  000402                      BR      4$
11695  030350  073267  175260      2$:     ASHC    TEMP3,#2        ;ASHC REGISTER 2 BY THE CONTETNS OF TEMP3
11699  030354  106737  025626      4$:     MFPS    @#PSWORD        ;SAVE PS
11703  030360  123737  025642  025626      CMPB    @#TEMP6,@#PSWORD;COMPARE PS WITH THE CONTENTS OF TEMP6
11704  030366  001401                      BEQ     11$
  (2)  030370  104000                      EMT                     ;WRONG PS
11705  030372  005237  025624      11$:    INC     @#COUNT
11706  030376  023702  025636              CMP     @#TEMP4,#2      ;IS THE RESULT IN R2 SAME AS TEMP4?
11707  030402  001401                      BEQ     12$
  (2)  030404  104000                      EMT                     ;WRONG RESULT IN R2
11708  030406  023703  025640      12$:    CMP     @#TEMP5,#3      ;IS THE RESULT IN R3 SAME AS TEMP5?
11709                                                               ;TEMP1 TEMP2 SHIFTED BY TEMP3-TEMP4 TEMP5
11710                                                               ;AND PS-TEMP6
11711  030412  001401                      BEQ     13$
  (2)  030414  104000                      EMT                     ;WRONG RESULT IN R1
11712  030416  021537  025624      13$:    CMP     (R5),@#COUNT    ;IS TEST NUMBER-COUNTER?
11713  030422  001401                      BEQ     14$
  (2)  030424  104000                      EMT                     ;NO
11714  030426  005215              14$:    INC     (R5)
11715  030430  021527  000160              CMP     (R5),#160       ;HAVE THE FIRST 159 TEST BEEN EXECUTED?
11716  030434  002014                      BGE     6$              ;YES
11717  030436  005237  025634              INC     @#TEMP3
11718  030442  000241                      CLC
11719  030444  006137  025640              ROL     @#TEMP5         ;ROTATE TEMP5 LEFT BY 1 PLACE
11720  030450  006137  025636              ROL     @#TEMP4         ;INTRODUCE CARRY FROM TEMP5 IN TEMP4
11721  030454  021527  000121              CMP     (R5),#121       ;IS IT TEST 121?
11722  030460  001004                      BNE     REG45
11723  030462  004467  000170              JSR     R4,RITSH        ;IF SO THEN GO AND INITIATE RIGHT SHIFT
11724  030466  004767  000220      6$:     JSR     #7,TST160
11725  030472  010501              REG45:  MOV     R5,R1           ;SAVE R5
11726  030474  013704  025630              MOV     @#TEMP1,#4      ;PLACE THE CONTENTS OF TEMP1 IN REGISTER 4
11727  030500  013705  025632              MOV     @#TEMP2,#4!1    ;PLACE THE CONTENTS OF TEMP2 IN REGISTER 5
11728  030504  000241                      CLC
11729  030506  032737  000001  001006      BIT     #1,@#$PASS      ;IS IT AN EVEN PASS ?
11730  030514  001004                      BNE     2$              ;IF NOT THEN GO TO 2$
11731  030516  013700  025634              MOV     @#TEMP3,R0      ;OTHERWISE EXECUTE ASHC INSTRUCTION IN MODE 0
11732  030522  073400                      ASHC    R0,R4           ;USING R4
11733  030524  000402                      BR      4$
11734  030526  073467  175102      2$:     ASHC    TEMP3,#4        ;ASHC REGISTER 4 BY THE CONTENTS OF TEMP3
11738  030532  106737  025626      4$:     MFPS    @#PSWORD        ;SAVE PS
11742  030536  123737  025642  025626      CMPB    @#TEMP6,@#PSWORD;COMPARE PS WITH THE CONTENTS OF TEMP6
11743  030544  001401                      BEQ     11$
  (2)  030546  104000                      EMT                     ;WRONG PS
11744  030550  005237  025624      11$:    INC     @#COUNT
11745  030554  023704  025636              CMP     @#TEMP4,#4      ;IS THE RESULT IN R4 SAME AS TEMP4?
11746  030560  001401                      BEQ     12$
  (2)  030562  104000                      EMT                     ;WRONG RESULT IN R4
```

```
11747  030564  023705  025640          12$:    CMP     @#TEMP5,#5      ;IS THE RESULT IN R5 SAME AS TEMP5?
11748                                                                   ;TEMP1 TEMP2 SHIFTED BY TEMP3-TEMP4 TEMP5
11749                                                                   ;AND PS=TEMP6
11750  030570  001401                          BEQ     13$
  (2)  030572  104000                          EMT                     ;WRONG RESULT IN R5
11751  030574  021137  025624          13$:    CMP     (R1),@#COUNT    ;IS TEST NUMBER=COUNTER?
11752  030600  001401                          BEQ     14$
  (2)  030602  104000                          EMT                     ;NO
11753  030604  010105          14$:    MOV     R1,R5           ;RESTORE R5
11754  030606  005215                  INC     (R5)
11755  030610  021527  000160                CMP     (R5),#160       ;HAVE THE FIRST 159 TEST BEEN EXECUTED?
11756  030614  002014                  BGE     6$              ;YES
11757  030616  005237  025634                INC     @#TEMP3
11758  030622  000241                  CLC
11759  030624  006137  025640                ROL     @#TEMP5         ;ROTATE TEMP5 LEFT BY 1 PLACE
11760  030630  006137  025636                ROL     @#TEMP4         ;INTRODUCE CARRY FROM TEMP5 IN TEMP4
11761  030634  021527  000121                CMP     (R5),#121       ;IS IT TEST 121?
11762  030640  001004                  BNE     8$
11763  030642  004467  000010                JSR     R4,RITSH        ;IF SO THEN GO AND INITIATE RIGHT SHIFT
11764  030646  004767  000040          6$:     JSR     #7,TST160
11765  030652  000167  177260          8$:     JMP     REG01
11766  030656  022424          RITSH:  CMP     (R4)+,(R4)+     ;MAKE R4 POINT TO THE NEXT REG TAG
11767  030660  012737  040000  025630        MOV     #40000,@#TEMP1  ;TEMP1=4000
11768  030666  005037  025632                CLR     @#TEMP2         ;TEMP2=0
11769  030672  012737  177742  025634        MOV     #-30.,@#TEMP3   ;TEMP3=-30
11770  030700  005037  025636                CLR     @#TEMP4         ;TEMP4=0
11771  030704  005237  025640                INC     @#TEMP5         ;TEMP5=1
11772  030710  000204                  RTS     R4
11773  030712  021527  000160          TST160: CMP     (R5),#160       ;IS IT TEST 160
11774  030716  001010                  BNE     TST161          ;IF NOT THEN TRY TEST 161
11775  030720  005037  025630                CLR     @#TEMP1         ;0 0 SHIFTED BY 0
11776  030724  005037  025636                CLR     @#TEMP4         ;IS EQUAL TO 0 0
11777  030730  012737  000004  025642        MOV     #4,@#TEMP6      ;AND PS=4
11778  030736  000207                  RTS     #7
11779  030740  021527  000161          TST161: CMP     (R5),#161       ;IS IT TEST 161
11780  030744  001004                  BNE     TST162
11781  030746  012737  177746  025634        MOV     #-32,@#TEMP3    ;0 0 SHIFTED BY -32=0 0, PS=4
11782  030754  000207                  RTS     #7
11783  030756  021527  000162          TST162: CMP     (R5),#162       ;IS IT TEST 162
11784  030762  001004                  BNE     TST163          ;IF NOT THEN TRY TEST 163
11785  030764  012737  000032  025634        MOV     #32,@#TEMP3     ;0 0 SHIFTED BY 32=0 0, PS=4
11786  030772  000207                  RTS     #7
11787  030774  021527  000163          TST163: CMP     (R5),#163       ;IS IT TEST 163?
11788  031000  001016                  BNE     TST164          ;IF NOT THEN TRY TEST 164
11789  031002  012737  052525  025630        MOV     #52525,@#TEMP1  ;52525 0
11790  031010  012737  177760  025634        MOV     #-16.,@#TEMP3   ;SHIFTED BY -16.
11791  031016  005037  025636                CLR     @#TEMP4
11792  031022  012737  052525  025640        MOV     #52525,@#TEMP5  ;IS EQUAL TO 0 52525
11793  031030  005037  025642                CLR     @#TEMP6         ;AND PS = 0
11794  031034  000207                  RTS     #7
11795  031036  021527  000164          TST164: CMP     (R5),#164       ;IS IT TEST 164?
11796  031042  001014                  BNE     TST165          ;IF NOT THEN TRY TEST 165
11797  031044  012737  125252  025630        MOV     #125252,@#TEMP1 ;125252 0 SHIFTED BY -16.
11798  031052  005337  025636                DEC     @#TEMP4
11799  031056  012737  125252  025640        MOV     #125252,@#TEMP5 ;IS EQUAL TO -1 125252
11800  031064  012737  000010  025642        MOV     #10,@#TEMP6     ;AND PS=10
```

```
11801  031072  000207                          RTS     %7
11802  031074  021527  000165          TST165: CMP     (R5),#165       ;IS IT TEST 165?
11803  031100  001007                          BNE     TST166          ;IF NOT THEN TRY TEST 166
11804  031102  012737  177777  025630          MOV     #-1,@#TEMP1     ;-1 0 SHIFTED BY -16
11805  031110  012737  177777  025640          MOV     #-1,@#TEMP5     ;IS EQUAL TO -1 -1, AND PS=10
11806  031116  000207                          RTS     %7
11807  031120  021527  000166          TST166: CMP     (R5),#166       ;IS IT TEST 166?
11808  031124  001011                          BNE     TST167          ;IF NOT THEN TRY TEST 167
11809  031126  012737  100000  025630          MOV     #100000,@#TEMP1 ;100000 0
11810  031134  012737  177740  025634          MOV     #-32.,@#TEMP3   ;SHIFTED BY -32 IS EQUAL TO -1 -1
11811  031142  005237  025642                  INC     @#TEMP6         ;AND PS=11
11812  031146  000207                          RTS     %7
11813  031150  021527  000167          TST167: CMP     (R5),#167       ;IS IT TEST 167?
11814  031154  001014                          BNE     TST170          ;IF NOT THEN TRY TEST 170
```

```
 11816  031156  005037  025630                  CLR     @#TEMP1
 11817  031162  005337  025632                  DEC     @#TEMP2         ;0 -1
```

```
11819  031166  012737  000020  025634          MOV     #16.,@#TEMP3    ;SHIFTED BY 16.
11820  031174  005037  025640                  CLR     @#TEMP5         ;IS EQUAL TO -1 0
11821  031200  005237  025642                  INC     @#TEMP6         ;AND PS=12
11822  031204  000207                          RTS     PC
11823  031206  021527  000170          TST170: CMP     (R5),#170       ;IS IT TEST 170?
```

```
 11825  031212  001007                    BNE     TST171          ;IF NOT THEN TRY TEST 171
```

```
11827  031214  012737  125252  025632          MOV     #125252,@#TEMP2 ;0 125252 SHIFTED BY 16
11828  031222  012737  125252  025636          MOV     #125252,@#TEMP4 ;IS EQUAL TO 125252 0, AND PS=12
11829  031230  000207                          RTS     %7
11830  031232  021527  000171          TST171: CMP     (R5),#171       ;IS IT TEST 171?
11831  031236  001010                          BNE     TST172          ;IF NOT THEN TRY TEST 172
11832  031240  005337  025634                  DEC     @#TEMP3         ;0 125252 SHIFTED BY 15
11833  031244  012737  052525  025636          MOV     #52525,@#TEMP4  ;IS EQUAL TO 52525 0
11834  031252  005037  025642                  CLR     @#TEMP6         ;AND PS=0
11835  031256  000207                          RTS     %7
11836  031260  021527  000172          TST172: CMP     (R5),#172       ;IS IT TEST 172?
11837  031264  001006                          BNE     TST173          ;IF NOT THEN TRY TEST 173
11838  031266  012737  052525  025632          MOV     #52525,@#TEMP2  ;0 52525
11839  031274  005237  025634                  INC     @#TEMP3         ;SHIFTED BY 16. IS EQUAL TO 52525 0, AND PS=0
11840  031300  000207                          RTS     %7
11841  031302  021527  000173          TST173: CMP     (R5),#173       ;IS IT TEST 173?
11842  031306  001014                          BNE     TST174          ;IF NOT THEN TRY TEST 174
11843  031310  012737  177777  025632          MOV     #-1,@#TEMP2     ;0 -1
11844  031316  005337  025634                  DEC     @#TEMP3         ;SHIFTED BY 15.
11845  031322  012737  077777  025636          MOV     #77777,@#TEMP4
11846  031330  012737  100000  025640          MOV     #100000,@#TEMP5 ;IS EQUAL TO 77777 100000, AND PS=0
11847  031336  000207                          RTS     %7
11848  031340  021527  000174          TST174: CMP     (R5),#174       ;IS IT TEST 174?
11849  031344  001013                          BNE     TST175          ;IF NOT THEN TRY TEST 175
11850  031346  012737  100000  025630          MOV     #100000,@#TEMP1
11851  031354  005337  025632                  DEC     @#TEMP2         ;100000 -2 SHIFTED BY 15.
11852  031360  005037  025640                  CLR     @#TEMP5         ;IS EQUAL TO 77777 0
11853  031364  012737  000002  025642          MOV     #2,@#TEMP6      ;AND PS=2
11854  031372  000207                          RTS     %7
11855  031374  021527  000175          TST175: CMP     (R5),#175       ;IS IT TEST 175?
11856  031400  001015                          BNE     ENT176          ;IF NOT THEN TRY TEST 176
11857  031402  012737  177777  025630          MOV     #-1,@#TEMP1
11858  031410  005037  025632                  CLR     @#TEMP2         ;-1 0
11859  031414  005237  025634                  INC     @#TEMP3         ;SHIFTED BY 16.
11860  031420  005037  025636                  CLR     @#TEMP4         ;IS EQUAL TO 0 0
11861  031424  012737  000007  025642          MOV     #7,@#TEMP6      ;AND PS=7
```

```
                                        RTS     %7
11863   031432  000207
11864   031434  021527  000176  ENT176: CMP     (R5),#176       ;IS THE PROGRM ENTERING TEST 176?
11865   031440  001401                  BEQ     1$
  (2)   031442  104000                  EMT                     ;TEST NUMBER GOOFED
11866
11867   031444  005726          1$:     TST     (SP)+           ;RESTORE STACK POINTER
11868
11869                           ;*****************************************************************
  (1)                           ;TEST:176       1 SHIFTED BY 8. = 400  PS = 0
  (1)                           ;*****************************************************************
  (1)
  (1)   031446                  TST176:
  (1)   031446  012701  000000          MOV     #DUMMY,%1       ;LOAD R1 WITH DUMMY
  (1)   031452  012701  000001          MOV     #1,%1!1         ;LOAD R1!1 WITH 1
  (1)   031456  000241                  CLC
  (1)   031460  073127  000010          ASHC    #8.,%1          ;SHIFT R1,R1!1 BY 8.
  (1)   031464  106737  025626          MFPS    @#PSWORD        ;SAVE PS
  (1)   031470  122737  000000  025626  CMPB    #0,@#PSWORD     ;IS THE PS 0?
  (2)   031476  001401                  BEQ     11$
  (3)   031500  104000                  EMT                     ;THE PS IS NOT EQUAL TO 0
  (1)   031502  022701  000400  11$:     CMP     #400,%1         ;IS THE RESULT 400?
  (2)   031506  001401                  BEQ     13$
  (3)   031510  104000                  EMT                     ;R1 IS NOT EQUAL TO 400
  (1)   031512                  13$:
  (1)   031512  005215                  INC     (R5)
  (1)
  (1)
11870                           ;*****************************************************************
  (1)                           ;TEST:177       -1 SHIFTED BY 15. = 100000  PS = 11
  (1)                           ;*****************************************************************
  (1)
  (1)
  (1)   031514                  TST177:
  (1)   031514  012703  000000          MOV     #DUMMY,%3       ;LOAD R3 WITH DUMMY
  (1)   031520  012703  177777          MOV     #-1,%3!1        ;LOAD R3!1 WITH -1
  (1)   031524  000241                  CLC
  (1)   031526  073327  000017          ASHC    #15.,%3         ;SHIFT R3,R3!1 BY 15.
  (1)   031532  106737  025626          MFPS    @#PSWORD        ;SAVE PS
  (1)   031536  122737  000011  025626  CMPB    #11,@#PSWORD    ;IS THE PS 11?
  (2)   031544  001401                  BEQ     11$
  (3)   031546  104000                  EMT                     ;THE PS IS NOT EQUAL TO 11
  (1)   031550  022703  100000  11$:     CMP     #100000,%3      ;IS THE RESULT 100000?
  (2)   031554  001401                  BEQ     13$
  (3)   031556  104000                  EMT                     ;R3 IS NOT EQUAL TO 100000
  (1)   031560                  13$:
  (1)   031560  005215                  INC     (R5)
  (1)
  (1)
11871                           ;*****************************************************************
  (1)                           ;TEST:200       52525 SHIFTED BY 0 = 52525  PS = 0
  (1)                           ;*****************************************************************
  (1)
  (1)   031562                  TST200:
  (1)   031562  010501                  MOV     R5,R1           ;SAVE R5
  (1)   031564  012705  000000          MOV     #DUMMY,%5       ;LOAD R5 WITH DUMMY
  (1)   031570  012705  052525          MOV     #52525,%5!1     ;LOAD R5!1 WITH 52525
  (1)   031574  000241                  CLC
```

```
  (1)  031576  073527  000000              ASHC    #0,%5           ;SHIFT R5,R5!1 BY 0
  (1)  031602  106737  025626              MFPS    @#PSWORD        ;SAVE PS
  (1)  031606  122737  000000  025626      CMPB    #0,@#PSWORD     ;IS THE PS 0?
  (2)  031614  001401                      BEQ     11$
  (3)  031616  104000                      EMT                     ;THE PS IS NOT EQUAL TO 0
  (1)  031620  022705  052525      11$:    CMP     #52525,%5       ;IS THE RESULT 52525?
  (2)  031624  001401                      BEQ     13$
  (3)  031626  104000                      EMT                     ;R5 IS NOT EQUAL TO 52525
  (1)  031630                      13$:
  (1)  031630  010105                      MOV     R1,R5           ;RESTORE R5
  (1)  031632  005215                      INC     (R5)
  (1)
  (1)
11872                              ;*****************************************************************************
  (1)                              ;TEST:201       20010 SHIFTED BY -13. = 101  PS = 0
  (1)                              ;*****************************************************************************
  (1)
  (1)  031634                      TST201:
  (1)  031634  012701  000000              MOV     #DUMMY,%1       ;LOAD R1 WITH DUMMY
  (1)  031640  012701  020010              MOV     #20010,%1!1     ;LOAD R1!1 WITH 20010
  (1)  031644  000241                      CLC
  (1)  031646  073127  177763              ASHC    #-13.,%1        ;SHIFT R1,R1!1 BY -13.
  (1)  031652  106737  025626              MFPS    @#PSWORD        ;SAVE PS
  (1)  031656  122737  000000  025626      CMPB    #0,@#PSWORD     ;IS THE PS 0?
  (2)  031664  001401                      BEQ     11$
  (3)  031666  104000                      EMT                     ;THE PS IS NOT EQUAL TO 0
  (1)  031670  022701  000101      11$:    CMP     #101,%1         ;IS THE RESULT 101?
  (2)  031674  001401                      BEQ     13$
  (3)  031676  104000                      EMT                     ;R1 IS NOT EQUAL TO 101
  (1)  031700                      13$:
  (1)  031700  005215                      INC     (R5)
  (1)
  (1)
11873                              ;*****************************************************************************
  (1)                              ;TEST:202       -1 SHIFTED BY 16. = 0  PS = 11
  (1)                              ;*****************************************************************************
  (1)
  (1)  031702                      TST202:
  (1)  031702  012703  000000              MOV     #DUMMY,%3       ;LOAD R3 WITH DUMMY
  (1)  031706  012703  177777              MOV     #-1,%3!1        ;LOAD R3!1 WITH -1
  (1)  031712  000241                      CLC
  (1)  031714  073327  000020              ASHC    #16.,%3         ;SHIFT R3,R3!1 BY 16.
  (1)  031720  106737  025626              MFPS    @#PSWORD        ;SAVE PS
  (1)  031724  122737  000011  025626      CMPB    #11,@#PSWORD    ;IS THE PS 11?
  (2)  031732  001401                      BEQ     11$
  (3)  031734  104000                      EMT                     ;THE PS IS NOT EQUAL TO 11
  (1)  031736  022703  000000      11$:    CMP     #0,%3           ;IS THE RESULT 0?
  (2)  031742  001401                      BEQ     13$
  (3)  031744  104000                      EMT                     ;R3 IS NOT EQUAL TO 0
  (1)  031746                      13$:
  (1)  031746  005215                      INC     (R5)
  (1)
  (1)
11874                              ;*****************************************************************************
  (1)                              ;TEST:203       1 SHIFTED BY -1 = 100000  PS = 1
  (1)                              ;*****************************************************************************
```

```
  (1)
  (1)  031750                          TST203:
  (1)  031750  010501                          MOV     R5,R1           ;SAVE R5
  (1)  031752  012705  000000                  MOV     #DUMMY,%5       ;LOAD R5 WITH DUMMY
  (1)  031756  012705  000001                  MOV     #1,%5!1         ;LOAD R5!1 WITH 1
  (1)  031762  000241                          CLC
  (1)  031764  073527  177777                  ASHC    #-1,%5          ;SHIFT R5,R5!1 BY -1
  (1)  031770  106737  025626                  MFPS    @#PSWORD        ;SAVE PS
  (1)  031774  122737  000001  025626          CMPB    #1,@#PSWORD     ;IS THE PS 1?
  (2)  032002  001401                          BEQ     11$
  (3)  032004  104000                          EMT                     ;THE PS IS NOT EQUAL TO 1
  (1)  032006  022705  100000          11$:    CMP     #100000,%5      ;IS THE RESULT 100000?
  (2)  032012  001401                          BEQ     13$
  (3)  032014  104000                          EMT                     ;R5 IS NOT EQUAL TO 100000
  (1)  032016                          13$:
  (1)  032016  010105                          MOV     R1,R5           ;RESTORE R5
  (1)  032020  005215                          INC     (R5)
  (1)
  (1)
11875                                  ;*******************************************************************
  (1)                                  ;TEST:204       125252 SHIFTED BY -16. = 125252  PS = 11
  (1)                                  ;*******************************************************************
  (1)
  (1)  032022                          TST204:
  (1)  032022  012701  000000                  MOV     #DUMMY,%1       ;LOAD R1 WITH DUMMY
  (1)  032026  012701  125252                  MOV     #125252,%1!1    ;LOAD R1!1 WITH 125252
  (1)  032032  000241                          CLC
  (1)  032034  073127  177760                  ASHC    #-16.,%1        ;SHIFT R1,R1!1 BY -16.
  (1)  032040  106737  025626                  MFPS    @#PSWORD        ;SAVE PS
  (1)  032044  122737  000011  025626          CMPB    #11,@#PSWORD    ;IS THE PS 11?
  (2)  032052  001401                          BEQ     11$
  (3)  032054  104000                          EMT                     ;THE PS IS NOT EQUAL TO 11
  (1)  032056  022701  125252          11$:    CMP     #125252,%1      ;IS THE RESULT 125252?
  (2)  032062  001401                          BEQ     13$
  (3)  032064  104000                          EMT                     ;R1 IS NOT EQUAL TO 125252
  (1)  032066                          13$:
  (1)  032066  005215                          INC     (R5)
  (1)
  (1)
11876                                  ;*******************************************************************
  (1)                                  ;TEST:205       125252 125252 SHIFTED BY 21. = 52500 000000  PS = 3
  (1)                                  ;*******************************************************************
  (1)
  (1)  032070                          TST205:
  (1)  032070  012702  125252                  MOV     #125252,%2      ;LOAD R2 WITH 125252
  (1)  032074  012703  125252                  MOV     #125252,%2!1    ;LOAD R2!1 WITH 125252
  (1)  032100  000241                          CLC
  (1)  032102  073227  000025                  ASHC    #21.,%2         ;SHIFT R2,R2!1 BY 21.
  (1)  032106  106737  025626                  MFPS    @#PSWORD        ;SAVE PS
  (1)  032112  122737  000003  025626          CMPB    #3,@#PSWORD     ;IS THE PS 3?
  (2)  032120  001401                          BEQ     11$
  (3)  032122  104000                          EMT                     ;THE PS IS NOT EQUAL TO 3
  (1)  032124  022702  052500          11$:    CMP     #52500,%2       ;IS THE RESULT 52500?
  (2)  032130  001401                          BEQ     12$
  (3)  032132  104000                          EMT                     ;R2 IS NOT EQUAL TO 52500
  (1)  032134  022703  000000          12$:    CMP     #000000,%2!1    ;IS THE RESULT 000000?
```

```
  (2)  032140  001401                    BEQ     13$
  (3)  032142  104000                    EMT                     ;R2!1 IS NOT EQUAL TO 000000
  (1)  032144                         13$:
  (1)  032144  005215                    INC     (R5)
  (1)
  (1)
11877
11878  032146  012702  177771            MOV     #-7,%2
11879  032152  012703  025644            MOV     #S1,%3
11880  032156  012704  025646            MOV     #S2,%4
11881
11882                                  ;*************************************************************************
  (1)                                  ;TEST:206       125252 125252 SHIFTED BY S1 = 177525 52525  PS = 10
  (1)                                  ;*************************************************************************
  (1)
  (1)  032162                          TST206:
  (1)  032162  012700  125252            MOV     #125252,%0              ;LOAD R0 WITH 125252
  (1)  032166  012701  125252            MOV     #125252,%0!1    ;LOAD R0!1 WITH 125252
  (1)  032172  000241                    CLC
  (1)  032174  073067  173444            ASHC    S1,%0           ;SHIFT R0,R0!1 BY S1
  (1)  032200  106737  025626            MFPS    @#PSWORD        ;SAVE PS
  (1)  032204  122737  000010  025626    CMPB    #10,@#PSWORD    ;IS THE PS 10?
  (2)  032212  001401                    BEQ     11$
  (3)  032214  104000                    EMT                     ;THE PS IS NOT EQUAL TO 10
  (1)  032216  022700  177525        11$:CMP     #177525,%0      ;IS THE RESULT 177525?
  (2)  032222  001401                    BEQ     12$
  (3)  032224  104000                    EMT                     ;R0 IS NOT EQUAL TO 177525
  (1)  032226  022701  052525        12$:CMP     #52525,%0!1     ;IS THE RESULT 52525?
  (2)  032232  001401                    BEQ     13$
  (3)  032234  104000                    EMT                     ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
  (1)  032236                         13$:
  (1)  032236  005215                    INC     (R5)
  (1)
  (1)
11883                                  ;*************************************************************************
  (1)                                  ;TEST:207       125252 125252 SHIFTED BY @S2 = 177525 52525  PS = 10
  (1)                                  ;*************************************************************************
  (1)
  (1)  032240                          TST207:
  (1)  032240  012700  125252            MOV     #125252,%0              ;LOAD R0 WITH 125252
  (1)  032244  012701  125252            MOV     #125252,%0!1    ;LOAD R0!1 WITH 125252
  (1)  032250  000241                    CLC
  (1)  032252  073077  173370            ASHC    @S2,%0          ;SHIFT R0,R0!1 BY @S2
  (1)  032256  106737  025626            MFPS    @#PSWORD        ;SAVE PS
  (1)  032262  122737  000010  025626    CMPB    #10,@#PSWORD    ;IS THE PS 10?
  (2)  032270  001401                    BEQ     11$
  (3)  032272  104000                    EMT                     ;THE PS IS NOT EQUAL TO 10
  (1)  032274  022700  177525        11$:CMP     #177525,%0      ;IS THE RESULT 177525?
  (2)  032300  001401                    BEQ     12$
  (3)  032302  104000                    EMT                     ;R0 IS NOT EQUAL TO 177525
  (1)  032304  022701  052525        12$:CMP     #52525,%0!1     ;IS THE RESULT 52525?
  (2)  032310  001401                    BEQ     13$
  (3)  032312  104000                    EMT                     ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
  (1)  032314                         13$:
  (1)  032314  005215                    INC     (R5)
  (1)
```

```
 (1)
11884                                   ;*******************************************************************
 (1)                                    ;TEST:210       125252 125252 SHIFTED BY @#S1 = 177525 52525  PS = 10
 (1)                                    ;*******************************************************************
 (1)
 (1)  032316                            TST210:
 (1)  032316  012700  125252                    MOV     #125252,#0              ;LOAD R0 WITH 125252
 (1)  032322  012701  125252                    MOV     #125252,#0!1    ;LOAD R0!1 WITH 125252
 (1)  032326  000241                            CLC
 (1)  032330  073037  025644                    ASHC    @#S1,#0         ;SHIFT R0,R0!1 BY @#S1
 (1)  032334  106737  025626                    MFPS    @#PSWORD        ;SAVE PS
 (1)  032340  122737  000010  025626            CMPB    #10,@#PSWORD    ;IS THE PS 10?
 (2)  032346  001401                            BEQ     11$
 (3)  032350  104000                            EMT                     ;THE PS IS NOT EQUAL TO 10
 (1)  032352  022700  177525            11$:    CMP     #177525,#0      ;IS THE RESULT 177525?
 (2)  032356  001401                            BEQ     12$
 (3)  032360  104000                            EMT                     ;R0 IS NOT EQUAL TO 177525
 (1)  032362  022701  052525            12$:    CMP     #52525,#0!1     ;IS THE RESULT 52525?
 (2)  032366  001401                            BEQ     13$
 (3)  032370  104000                            EMT                     ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
 (1)  032372                            13$:
 (1)  032372  005215                            INC     (R5)
 (1)
 (1)
11885                                   ;*******************************************************************
 (1)                                    ;TEST:211       125252 125252 SHIFTED BY (3) = 177525 52525  PS = 10
 (1)                                    ;*******************************************************************
 (1)
 (1)  032374                            TST211:
 (1)  032374  012700  125252                    MOV     #125252,#0              ;LOAD R0 WITH 125252
 (1)  032400  012701  125252                    MOV     #125252,#0!1    ;LOAD R0!1 WITH 125252
 (1)  032404  000241                            CLC
 (1)  032406  073013                            ASHC    (3),#0          ;SHIFT R0,R0!1 BY (3)
 (1)  032410  106737  025626                    MFPS    @#PSWORD        ;SAVE PS
 (1)  032414  122737  000010  025626            CMPB    #10,@#PSWORD    ;IS THE PS 10?
 (2)  032422  001401                            BEQ     11$
 (3)  032424  104000                            EMT                     ;THE PS IS NOT EQUAL TO 10
 (1)  032426  022700  177525            11$:    CMP     #177525,#0      ;IS THE RESULT 177525?
 (2)  032432  001401                            BEQ     12$
 (3)  032434  104000                            EMT                     ;R0 IS NOT EQUAL TO 177525
 (1)  032436  022701  052525            12$:    CMP     #52525,#0!1     ;IS THE RESULT 52525?
 (2)  032442  001401                            BEQ     13$
 (3)  032444  104000                            EMT                     ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
 (1)  032446                            13$:
 (1)  032446  005215                            INC     (R5)
 (1)
 (1)
11886                                   ;*******************************************************************
 (1)                                    ;TEST:212       125252 125252 SHIFTED BY (3)+ = 177525 52525  PS = 10
 (1)                                    ;*******************************************************************
 (1)
 (1)  032450                            TST212:
 (1)  032450  012700  125252                    MOV     #125252,#0              ;LOAD R0 WITH 125252
 (1)  032454  012701  125252                    MOV     #125252,#0!1    ;LOAD R0!1 WITH 125252
 (1)  032460  000241                            CLC
 (1)  032462  073023                            ASHC    (3)+,#0         ;SHIFT R0,R0!1 BY (3)+
```

```
(1)  032464  106737  025626                MFPS    @#PSWORD        ;SAVE PS
(1)  032470  122737  000010  025626        CMPB    #10,@#PSWORD    ;IS THE PS 10?
(2)  032476  001401                        BEQ     11$
(3)  032500  104000                        EMT                     ;THE PS IS NOT EQUAL TO 10
(1)  032502  022700  177525        11$:    CMP     #177525,R0      ;IS THE RESULT 177525?
(2)  032506  001401                        BEQ     12$
(3)  032510  104000                        EMT                     ;R0 IS NOT EQUAL TO 177525
(1)  032512  022701  052525        12$:    CMP     #52525,R0+1     ;IS THE RESULT 52525?
(2)  032516  001401                        BEQ     13$
(3)  032520  104000                        EMT                     ;R0+1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
(1)  032522                        13$:
(1)  032522  005215                        INC     (R5)
(1)
(1)
11887                              ;*************************************************************************
(1)                                ;TEST:213       125252 125252 SHIFTED BY -(3) = 177525 52525  PS = 10
(1)                                ;*************************************************************************
(1)
(1)  032524                        TST213:
(1)  032524  012700  125252                MOV     #125252,R0              ;LOAD R0 WITH 125252
(1)  032530  012701  125252                MOV     #125252,R0+1    ;LOAD R0+1 WITH 125252
(1)  032534  000241                        CLC
(1)  032536  073043                        ASHC    -(3),R0         ;SHIFT R0,R0+1 BY -(3)
(1)  032540  106737  025626                MFPS    @#PSWORD        ;SAVE PS
(1)  032544  122737  000010  025626        CMPB    #10,@#PSWORD    ;IS THE PS 10?
(2)  032552  001401                        BEQ     11$
(3)  032554  104000                        EMT                     ;THE PS IS NOT EQUAL TO 10
(1)  032556  022700  177525        11$:    CMP     #177525,R0      ;IS THE RESULT 177525?
(2)  032562  001401                        BEQ     12$
(3)  032564  104000                        EMT                     ;R0 IS NOT EQUAL TO 177525
(1)  032566  022701  052525        12$:    CMP     #52525,R0+1     ;IS THE RESULT 52525?
(2)  032572  001401                        BEQ     13$
(3)  032574  104000                        EMT                     ;R0+1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
(1)  032576                        13$:
(1)  032576  005215                        INC     (R5)
(1)
(1)
11888                              ;*************************************************************************
(1)                                ;TEST:214       125252 125252 SHIFTED BY 2(4) = 177252 125252  PS = 11
(1)                                ;*************************************************************************
(1)
(1)  032600                        TST214:
(1)  032600  012700  125252                MOV     #125252,R0              ;LOAD R0 WITH 125252
(1)  032604  012701  125252                MOV     #125252,R0+1    ;LOAD R0+1 WITH 125252
(1)  032610  000241                        CLC
(1)  032612  073064  000002                ASHC    2(4),R0         ;SHIFT R0,R0+1 BY 2(4)
(1)  032616  106737  025626                MFPS    @#PSWORD        ;SAVE PS
(1)  032622  122737  000011  025626        CMPB    #11,@#PSWORD    ;IS THE PS 11?
(2)  032630  001401                        BEQ     11$
(3)  032632  104000                        EMT                     ;THE PS IS NOT EQUAL TO 11
(1)  032634  022700  177252        11$:    CMP     #177252,R0      ;IS THE RESULT 177252?
(2)  032640  001401                        BEQ     12$
(3)  032642  104000                        EMT                     ;R0 IS NOT EQUAL TO 177252
(1)  032644  022701  125252        12$:    CMP     #125252,R0+1    ;IS THE RESULT 125252?
(2)  032650  001401                        BEQ     13$
(3)  032652  104000                        EMT                     ;R0+1 IS NOT EQUAL TO 125252 OR INCORRECT SEQUENCE
```

```
  (1)  032654                          13$:
  (1)  032654  005215                          INC     (R5)
  (1)
  (1)
11889                                  ;*****************************************************************************
  (1)                                  ;TEST:215       125252 125252 SHIFTED BY @(4) = 177525 52525  PS = 10
  (1)                                  ;*****************************************************************************
  (1)
  (1)  032656                          TST215:
  (1)  032656  012700  125252                  MOV     #125252,%0              ;LOAD R0 WITH 125252
  (1)  032662  012701  125252                  MOV     #125252,%0!1    ;LOAD R0!1 WITH 125252
  (1)  032666  000241                          CLC
  (1)  032670  073074  000000                  ASHC    @(4),%0         ;SHIFT R0,R0!1 BY @(4)
  (1)  032674  106737  025626                  MFPS    @#PSWORD        ;SAVE PS
  (1)  032700  122737  000010  025626          CMPB    #10,@#PSWORD    ;IS THE PS 10?
  (2)  032706  001401                          BEQ     11$
  (3)  032710  104000                          EMT                     ;THE PS IS NOT EQUAL TO 10
  (1)  032712  022700  177525          11$:    CMP     #177525,%0      ;IS THE RESULT 177525?
  (2)  032716  001401                          BEQ     12$
  (3)  032720  104000                          EMT                     ;R0 IS NOT EQUAL TO 177525
  (1)  032722  022701  052525          12$:    CMP     #52525,%0!1     ;IS THE RESULT 52525?
  (2)  032726  001401                          BEQ     13$
  (3)  032730  104000                          EMT                     ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
  (1)  032732                          13$:
  (1)  032732  005215                          INC     (R5)
  (1)
  (1)
11890                                  ;*****************************************************************************
  (1)                                  ;TEST:216       125252 125252 SHIFTED BY @(4)+ = 177525 52525  PS = 10
  (1)                                  ;*****************************************************************************
  (1)
  (1)  032734                          TST216:
  (1)  032734  012700  125252                  MOV     #125252,%0              ;LOAD R0 WITH 125252
  (1)  032740  012701  125252                  MOV     #125252,%0!1    ;LOAD R0!1 WITH 125252
  (1)  032744  000241                          CLC
  (1)  032746  073034                          ASHC    @(4)+,%0                ;SHIFT R0,R0!1 BY @(4)+
  (1)  032750  106737  025626                  MFPS    @#PSWORD        ;SAVE PS
  (1)  032754  122737  000010  025626          CMPB    #10,@#PSWORD    ;IS THE PS 10?
  (2)  032762  001401                          BEQ     11$
  (3)  032764  104000                          EMT                     ;THE PS IS NOT EQUAL TO 10
  (1)  032766  022700  177525          11$:    CMP     #177525,%0      ;IS THE RESULT 177525?
  (2)  032772  001401                          BEQ     12$
  (3)  032774  104000                          EMT                     ;R0 IS NOT EQUAL TO 177525
  (1)  032776  022701  052525          12$:    CMP     #52525,%0!1     ;IS THE RESULT 52525?
  (2)  033002  001401                          BEQ     13$
  (3)  033004  104000                          EMT                     ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
  (1)  033006                          13$:
  (1)  033006  005215                          INC     (R5)
  (1)
  (1)
11891                                  ;*****************************************************************************
  (1)                                  ;TEST:217       125252 125252 SHIFTED BY @-(4) = 177525 52525  PS = 10
  (1)                                  ;*****************************************************************************
  (1)
  (1)  033010                          TST217:
  (1)  033010  012700  125252                  MOV     #125252,%0              ;LOAD R0 WITH 125252
```

```
  (1)  033014  012701  125252              MOV     #125252,%0!1    ;LOAD R0!1 WITH 125252
  (1)  033020  000241                      CLC
  (1)  033022  073054                      ASHC    @-(4),%0                ;SHIFT R0,R0!1 BY @-(4)
  (1)  033024  106737  025626              MFPS    @#PSWORD        ;SAVE PS
  (1)  033030  122737  000010  025626      CMPB    #10,@#PSWORD    ;IS THE PS 10?
  (2)  033036  001401                      BEQ     11$
  (3)  033040  104000                      EMT                     ;THE PS IS NOT EQUAL TO 10
  (1)  033042  022700  177525      11$:    CMP     #177525,%0      ;IS THE RESULT 177525?
  (2)  033046  001401                      BEQ     12$
  (3)  033050  104000                      EMT                     ;R0 IS NOT EQUAL TO 177525
  (1)  033052  022701  052525      12$:    CMP     #52525,%0!1     ;IS THE RESULT 52525?
  (2)  033056  001401                      BEQ     13$
  (3)  033060  104000                      EMT                     ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
  (1)  033062                      13$:
  (1)  033062  005215                      INC     (R5)
  (1)
  (1)
11892
11893
11894
11895
11896
11897
11898
```

```
11981                                   ;*********************************
11982                                   ;     MUL INSTRUCTION TESTS
11983                                   ;*********************************
11984
11985
11986
11987
11988                                   ;*****************************************************************************
  (1)                                   ;TEST:220       MUL       1 * #0 = 0 0         PS = 4
  (1)                                   ;*****************************************************************************
  (1)
  (1)   033064                          TST220:
  (1)   033064  012700  000001                 MOV     #1,%0           ;LOAD MULTIPLICAND WITH 1
  (1)   033070  070027  000000                 MUL     #0,%0           ;MULTIPLY 1 * #0
  (1)   033074  106737  025626                 MFPS    @#PSWORD        ;SAVE PS
  (1)   033100  122737  000004  025626         CMPB    #4,@#PSWORD     ;IS PS = 4
  (2)   033106  001401                         BEQ     11$
  (3)   033110  104000                         EMT                     ;PS IS WRONG
  (1)   033112  022700  000000         11$:    CMP     #0,%0           ;IS HIGH ORDER = 0
  (2)   033116  001401                         BEQ     12$
  (3)   033120  104000                         EMT                     ;HIGH ORDER IS WRONG
  (1)   033122  022701  000000         12$:    CMP     #0,%0!1         ;IS LOW ORDER = 0
  (2)   033126  001401                         BEQ     13$
  (3)   033130  104000                         EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
  (1)   033132                          13$:
  (1)   033132  005215                         INC     (R5)
  (1)
  (1)
11989                                   ;*****************************************************************************
  (1)                                   ;TEST:221       MUL       -1 * #1 = -1 -1        PS = 10
  (1)                                   ;*****************************************************************************
  (1)
  (1)   033134                          TST221:
  (1)   033134  012700  177777                 MOV     #-1,%0          ;LOAD MULTIPLICAND WITH -1
  (1)   033140  070027  000001                 MUL     #1,%0           ;MULTIPLY -1 * #1
  (1)   033144  106737  025626                 MFPS    @#PSWORD        ;SAVE PS
  (1)   033150  122737  000010  025626         CMPB    #10,@#PSWORD    ;IS PS = 10
  (2)   033156  001401                         BEQ     11$
  (3)   033160  104000                         EMT                     ;PS IS WRONG
  (1)   033162  022700  177777         11$:    CMP     #-1,%0          ;IS HIGH ORDER = -1
  (2)   033166  001401                         BEQ     12$
  (3)   033170  104000                         EMT                     ;HIGH ORDER IS WRONG
  (1)   033172  022701  177777         12$:    CMP     #-1,%0!1        ;IS LOW ORDER = -1
  (2)   033176  001401                         BEQ     13$
  (3)   033200  104000                         EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
  (1)   033202                          13$:
  (1)   033202  005215                         INC     (R5)
  (1)
  (1)
11990                                   ;*****************************************************************************
  (1)                                   ;TEST:222       MUL       2 * #2 = 0 4         PS = 0
  (1)                                   ;*****************************************************************************
  (1)
  (1)   033204                          TST222:
  (1)   033204  012702  000002                 MOV     #2,%2           ;LOAD MULTIPLICAND WITH 2
  (1)   033210  070227  000002                 MUL     #2,%2           ;MULTIPLY 2 * #2
```

```
  (1)  033214  106737  025626                  MFPS    @#PSWORD          ;SAVE PS
  (1)  033220  122737  000000  025626          CMPB    #0,@#PSWORD       ;IS PS = 0
  (2)  033226  001401                          BEQ     11$
  (3)  033230  104000                          EMT                       ;PS IS WRONG
  (1)  033232  022702  000000          11$:    CMP     #0,%2             ;IS HIGH ORDER = 0
  (2)  033236  001401                          BEQ     12$
  (3)  033240  104000                          EMT                       ;HIGH ORDER IS WRONG
  (1)  033242  022703  000004          12$:    CMP     #4,%2!1           ;IS LOW ORDER = 4
  (2)  033246  001401                          BEQ     13$
  (3)  033250  104000                          EMT                       ;LOW ORDER IS WRONG OR WRONG SEQUENCE
  (1)  033252                          13$:
  (1)  033252  005215                          INC     (R5)
  (1)
  (1)
11991                                  ;*****************************************************************
  (1)                                  ;TEST:223       MUL       1000 * #200 = 1 0        PS = 1
  (1)                                  ;*****************************************************************
  (1)
  (1)  033254                          TST223:
  (1)  033254  010501                          MOV     R5,R1             ;SAVE R5
  (1)  033256  012704  001000                  MOV     #1000,%4          ;LOAD MULTIPLICAND WITH 1000
  (1)  033262  070427  000200                  MUL     #200,%4           ;MULTIPLY 1000 * #200
  (1)  033266  106737  025626                  MFPS    @#PSWORD          ;SAVE PS
  (1)  033272  122737  000001  025626          CMPB    #1,@#PSWORD       ;IS PS = 1
  (2)  033300  001401                          BEQ     11$
  (3)  033302  104000                          EMT                       ;PS IS WRONG
  (1)  033304  022704  000001          11$:    CMP     #1,%4             ;IS HIGH ORDER = 1
  (2)  033310  001401                          BEQ     12$
  (3)  033312  104000                          EMT                       ;HIGH ORDER IS WRONG
  (1)  033314  022705  000000          12$:    CMP     #0,%4!1           ;IS LOW ORDER = 0
  (2)  033320  001401                          BEQ     13$
  (3)  033322  104000                          EMT                       ;LOW ORDER IS WRONG OR WRONG SEQUENCE
  (1)  033324                          13$:
  (1)  033324  010105                          MOV     R1,R5             ;RESTORE R5
  (1)  033326  005215                          INC     (R5)
  (1)
  (1)
11992                                  ;*****************************************************************
  (1)                                  ;TEST:224       MUL       2 * #77777 = 0 177776        PS = 1
  (1)                                  ;*****************************************************************
  (1)
  (1)  033330                          TST224:
  (1)  033330  012700  000002                  MOV     #2,%0             ;LOAD MULTIPLICAND WITH 2
  (1)  033334  070027  077777                  MUL     #77777,%0         ;MULTIPLY 2 * #77777
  (1)  033340  106737  025626                  MFPS    @#PSWORD          ;SAVE PS
  (1)  033344  122737  000001  025626          CMPB    #1,@#PSWORD       ;IS PS = 1
  (2)  033352  001401                          BEQ     11$
  (3)  033354  104000                          EMT                       ;PS IS WRONG
  (1)  033356  022700  000000          11$:    CMP     #0,%0             ;IS HIGH ORDER = 0
  (2)  033362  001401                          BEQ     12$
  (3)  033364  104000                          EMT                       ;HIGH ORDER IS WRONG
  (1)  033366  022701  177776          12$:    CMP     #177776,%0!1      ;IS LOW ORDER = 177776
  (2)  033372  001401                          BEQ     13$
  (3)  033374  104000                          EMT                       ;LOW ORDER IS WRONG OR WRONG SEQUENCE
  (1)  033376                          13$:
  (1)  033376  005215                          INC     (R5)
```

```
  (1)
  (1)
11993                                   ;*****************************************************************
  (1)                                   ;TEST:225       MUL       7777 * #10 = 0 77770        PS = 0
  (1)                                   ;*****************************************************************
  (1)
  (1)  033400                           TST225:
  (1)  033400  012702  007777                   MOV     #7777,%2          ;LOAD MULTIPLICAND WITH 7777
  (1)  033404  070227  000010                   MUL     #10,%2            ;MULTIPLY 7777 * #10
  (1)  033410  106737  025626                   MFPS    @#PSWORD          ;SAVE PS
  (1)  033414  122737  000000  025626           CMPB    #0,@#PSWORD       ;IS PS = 0
  (2)  033422  001401                           BEQ     11$
  (3)  033424  104000                           EMT                       ;PS IS WRONG
  (1)  033426  022702  000000           11$:    CMP     #0,%2             ;IS HIGH ORDER = 0
  (2)  033432  001401                           BEQ     12$
  (3)  033434  104000                           EMT                       ;HIGH ORDER IS WRONG
  (1)  033436  022703  077770           12$:    CMP     #77770,%2!1       ;IS LOW ORDER = 77770
  (2)  033442  001401                           BEQ     13$
  (3)  033444  104000                           EMT                       ;LOW ORDER IS WRONG OR WRONG SEQUENCE
  (1)  033446                           13$:
  (1)  033446  005215                           INC     (R5)
  (1)
  (1)
11994                                   ;*****************************************************************
  (1)                                   ;TEST:226       MUL       77777 * #77777 = 37777 1        PS = 1
  (1)                                   ;*****************************************************************
  (1)
  (1)  033450                           TST226:
  (1)  033450  010501                           MOV     R5,R1             ;SAVE R5
  (1)  033452  012704  077777                   MOV     #77777,%4         ;LOAD MULTIPLICAND WITH 77777
  (1)  033456  070427  077777                   MUL     #77777,%4         ;MULTIPLY 77777 * #77777
  (1)  033462  106737  025626                   MFPS    @#PSWORD          ;SAVE PS
  (1)  033466  122737  000001  025626           CMPB    #1,@#PSWORD       ;IS PS = 1
  (2)  033474  001401                           BEQ     11$
  (3)  033476  104000                           EMT                       ;PS IS WRONG
  (1)  033500  022704  037777           11$:    CMP     #37777,%4         ;IS HIGH ORDER = 37777
  (2)  033504  001401                           BEQ     12$
  (3)  033506  104000                           EMT                       ;HIGH ORDER IS WRONG
  (1)  033510  022705  000001           12$:    CMP     #1,%4!1           ;IS LOW ORDER = 1
  (2)  033514  001401                           BEQ     13$
  (3)  033516  104000                           EMT                       ;LOW ORDER IS WRONG OR WRONG SEQUENCE
  (1)  033520                           13$:
  (1)  033520  010105                           MOV     R1,R5             ;RESTORE R5
  (1)  033522  005215                           INC     (R5)
  (1)
  (1)
11995                                   ;*****************************************************************
  (1)                                   ;TEST:227       MUL       -1 * #77777 = -1 100001        PS = 10
  (1)                                   ;*****************************************************************
  (1)
  (1)  033524                           TST227:
  (1)  033524  012702  177777                   MOV     #-1,%2            ;LOAD MULTIPLICAND WITH -1
  (1)  033530  070227  077777                   MUL     #77777,%2         ;MULTIPLY -1 * #77777
  (1)  033534  106737  025626                   MFPS    @#PSWORD          ;SAVE PS
  (1)  033540  122737  000010  025626           CMPB    #10,@#PSWORD      ;IS PS = 10
  (2)  033546  001401                           BEQ     11$
```

```
  (3)  033550  104000                          EMT                     ;PS IS WRONG
  (1)  033552  022702  177777          11$:    CMP     #-1,%2          ;IS HIGH ORDER = -1
  (2)  033556  001401                          BEQ     12$
  (3)  033560  104000                          EMT                     ;HIGH ORDER IS WRONG
  (1)  033562  022703  100001          12$:    CMP     #100001,%2!1    ;IS LOW ORDER = 100001
  (2)  033566  001401                          BEQ     13$
  (3)  033570  104000                          EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
  (1)  033572                          13$:
  (1)  033572  005215                          INC     (R5)
  (1)
  (1)
11996                                  ;*******************************************************************
  (1)                                  ;TEST:230       MUL       -2 * #77777 = -1 2        PS = 11
  (1)                                  ;*******************************************************************
  (1)
  (1)  033574                          TST230:
  (1)  033574  012700  177776                  MOV     #-2,%0          ;LOAD MULTIPLICAND WITH -2
  (1)  033600  070027  077777                  MUL     #77777,%0       ;MULTIPLY -2 * #77777
  (1)  033604  106737  025626                  MFPS    @#PSWORD        ;SAVE PS
  (1)  033610  122737  000011  025626          CMPB    #11,@#PSWORD    ;IS PS = 11
  (2)  033616  001401                          BEQ     11$
  (3)  033620  104000                          EMT                     ;PS IS WRONG
  (1)  033622  022700  177777          11$:    CMP     #-1,%0          ;IS HIGH ORDER = -1
  (2)  033626  001401                          BEQ     12$
  (3)  033630  104000                          EMT                     ;HIGH ORDER IS WRONG
  (1)  033632  022701  000002          12$:    CMP     #2,%0!1         ;IS LOW ORDER = 2
  (2)  033636  001401                          BEQ     13$
  (3)  033640  104000                          EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
  (1)  033642                          13$:
  (1)  033642  005215                          INC     (R5)
  (1)
  (1)
11997                                  ;*******************************************************************
  (1)                                  ;TEST:231       MUL       125252 * #2 = -1 52524        PS = 11
  (1)                                  ;*******************************************************************
  (1)
  (1)  033644                          TST231:
  (1)  033644  012702  125252                  MOV     #125252,%2      ;LOAD MULTIPLICAND WITH 125252
  (1)  033650  070227  000002                  MUL     #2,%2           ;MULTIPLY 125252 * #2
  (1)  033654  106737  025626                  MFPS    @#PSWORD        ;SAVE PS
  (1)  033660  122737  000011  025626          CMPB    #11,@#PSWORD    ;IS PS = 11
  (2)  033666  001401                          BEQ     11$
  (3)  033670  104000                          EMT                     ;PS IS WRONG
  (1)  033672  022702  177777          11$:    CMP     #-1,%2          ;IS HIGH ORDER = -1
  (2)  033676  001401                          BEQ     12$
  (3)  033700  104000                          EMT                     ;HIGH ORDER IS WRONG
  (1)  033702  022703  052524          12$:    CMP     #52524,%2!1     ;IS LOW ORDER = 52524
  (2)  033706  001401                          BEQ     13$
  (3)  033710  104000                          EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
  (1)  033712                          13$:
  (1)  033712  005215                          INC     (R5)
  (1)
  (1)
11998                                  ;*******************************************************************
  (1)                                  ;TEST:232       MUL       125252 * #40000 = 165252 100000        PS = 11
  (1)                                  ;*******************************************************************
```

```
  (1)
  (1)  033714                          TST232:
  (1)  033714  010501                          MOV     R5,R1           ;SAVE R5
  (1)  033716  012704  125252                  MOV     #125252,%4      ;LOAD MULTIPLICAND WITH 125252
  (1)  033722  070427  040000                  MUL     #40000,%4       ;MULTIPLY 125252 * #40000
  (1)  033726  106737  025626                  MFPS    @#PSWORD        ;SAVE PS
  (1)  033732  122737  000011  025626          CMPB    #11,@#PSWORD    ;IS PS = 11
  (2)  033740  001401                          BEQ     11$
  (3)  033742  104000                          EMT                     ;PS IS WRONG
  (1)  033744  022704  165252          11$:    CMP     #165252,%4      ;IS HIGH ORDER = 165252
  (2)  033750  001401                          BEQ     12$
  (3)  033752  104000                          EMT                     ;HIGH ORDER IS WRONG
  (1)  033754  022705  100000          12$:    CMP     #100000,%4!1    ;IS LOW ORDER = 100000
  (2)  033760  001401                          BEQ     13$
  (3)  033762  104000                          EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
  (1)  033764                          13$:
  (1)  033764  010105                          MOV     R1,R5           ;RESTORE R5
  (1)  033766  005215                          INC     (R5)
  (1)
  (1)
11999                                  ;*****************************************************************************
  (1)                                  ;TEST:233       MUL       107070 * #107070 = 31222 26100        PS = 1
  (1)                                  ;*****************************************************************************
  (1)
  (1)  033770                          TST233:
  (1)  033770  012700  107070                  MOV     #107070,%0      ;LOAD MULTIPLICAND WITH 107070
  (1)  033774  070027  107070                  MUL     #107070,%0      ;MULTIPLY 107070 * #107070
  (1)  034000  106737  025626                  MFPS    @#PSWORD        ;SAVE PS
  (1)  034004  122737  000001  025626          CMPB    #1,@#PSWORD     ;IS PS = 1
  (2)  034012  001401                          BEQ     11$
  (3)  034014  104000                          EMT                     ;PS IS WRONG
  (1)  034016  022700  031222          11$:    CMP     #31222,%0       ;IS HIGH ORDER = 31222
  (2)  034022  001401                          BEQ     12$
  (3)  034024  104000                          EMT                     ;HIGH ORDER IS WRONG
  (1)  034026  022701  026100          12$:    CMP     #26100,%0!1     ;IS LOW ORDER = 26100
  (2)  034032  001401                          BEQ     13$
  (3)  034034  104000                          EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
  (1)  034036                          13$:
  (1)  034036  005215                          INC     (R5)
  (1)
  (1)
12000                                  ;*****************************************************************************
  (1)                                  ;TEST:234       MUL       -1 * #1 = -1 -1         PS = 10
  (1)                                  ;*****************************************************************************
  (1)
  (1)  034040                          TST234:
  (1)  034040  012701  177777                  MOV     #-1,%1          ;LOAD MULTIPLICAND WITH -1
  (1)  034044  070127  000001                  MUL     #1,%1           ;MULTIPLY -1 * #1
  (1)  034050  106737  025626                  MFPS    @#PSWORD        ;SAVE PS
  (1)  034054  122737  000010  025626          CMPB    #10,@#PSWORD    ;IS PS = 10
  (2)  034062  001401                          BEQ     11$
  (3)  034064  104000                          EMT                     ;PS IS WRONG
  (1)  034066  022701  177777          11$:    CMP     #-1,%1          ;IS HIGH ORDER = -1
  (2)  034072  001401                          BEQ     12$
  (3)  034074  104000                          EMT                     ;HIGH ORDER IS WRONG
  (1)  034076  022701  177777          12$:    CMP     #-1,%1!1        ;IS LOW ORDER = -1
```

```
  (2)  034102  001401                        BEQ     13$
  (3)  034104  104000                        EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
  (1)  034106                          13$:
  (1)  034106  005215                        INC     (R5)
  (1)
  (1)
12001                                 ;*************************************************************************
  (1)                                 ;TEST:235       MUL       -1 * #0 = 0 0         PS = 4
  (1)                                 ;*************************************************************************
  (1)
  (1)  034110                         TST235:
  (1)  034110  012703  177777                MOV     #-1,%3          ;LOAD MULTIPLICAND WITH -1
  (1)  034114  070327  000000                MUL     #0,%3           ;MULTIPLY -1 * #0
  (1)  034120  106737  025626                MFPS    @#PSWORD        ;SAVE PS
  (1)  034124  122737  000004  025626        CMPB    #4,@#PSWORD     ;IS PS = 4
  (2)  034132  001401                        BEQ     11$
  (3)  034134  104000                        EMT                     ;PS IS WRONG
  (1)  034136  022703  000000          11$:  CMP     #0,%3           ;IS HIGH ORDER = 0
  (2)  034142  001401                        BEQ     12$
  (3)  034144  104000                        EMT                     ;HIGH ORDER IS WRONG
  (1)  034146  022703  000000          12$:  CMP     #0,%3!1         ;IS LOW ORDER = 0
  (2)  034152  001401                        BEQ     13$
  (3)  034154  104000                        EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
  (1)  034156                          13$:
  (1)  034156  005215                        INC     (R5)
  (1)
  (1)
12002                                 ;*************************************************************************
  (1)                                 ;TEST:236       MUL       77777 * #100000 = 100000 100000       PS = 11
  (1)                                 ;*************************************************************************
  (1)
  (1)  034160                         TST236:
  (1)  034160  010501                        MOV     R5,R1           ;SAVE R5
  (1)  034162  012705  077777                MOV     #77777,%5       ;LOAD MULTIPLICAND WITH 77777
  (1)  034166  070527  100000                MUL     #100000,%5      ;MULTIPLY 77777 * #100000
  (1)  034172  106737  025626                MFPS    @#PSWORD        ;SAVE PS
  (1)  034176  122737  000011  025626        CMPB    #11,@#PSWORD    ;IS PS = 11
  (2)  034204  001401                        BEQ     11$
  (3)  034206  104000                        EMT                     ;PS IS WRONG
  (1)  034210  022705  100000          11$:  CMP     #100000,%5      ;IS HIGH ORDER = 100000
  (2)  034214  001401                        BEQ     12$
  (3)  034216  104000                        EMT                     ;HIGH ORDER IS WRONG
  (1)  034220  022705  100000          12$:  CMP     #100000,%5!1    ;IS LOW ORDER = 100000
  (2)  034224  001401                        BEQ     13$
  (3)  034226  104000                        EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
  (1)  034230                          13$:
  (1)  034230  010105                        MOV     R1,R5           ;RESTORE R5
  (1)  034232  005215                        INC     (R5)
  (1)
  (1)
12003                                 ;*************************************************************************
  (1)                                 ;TEST:237       MUL       -1 * #77777 = 100001 100001       PS = 10
  (1)                                 ;*************************************************************************
  (1)
  (1)  034234                         TST237:
  (1)  034234  012701  177777                MOV     #-1,%1          ;LOAD MULTIPLICAND WITH -1
```

```
  (1)  034240  070127  077777                MUL     #77777,%1       ;MULTIPLY -1 * #77777
  (1)  034244  106737  025626                MFPS    @#PSWORD        ;SAVE PS
  (1)  034250  122737  000010  025626        CMPB    #10,@#PSWORD    ;IS PS = 10
  (2)  034256  001401                        BEQ     11$
  (3)  034260  104000                        EMT                     ;PS IS WRONG
  (1)  034262  022701  100001        11$:    CMP     #100001,%1      ;IS HIGH ORDER = 100001
  (2)  034266  001401                        BEQ     12$
  (3)  034270  104000                        EMT                     ;HIGH ORDER IS WRONG
  (1)  034272  022701  100001        12$:    CMP     #100001,%1!1    ;IS LOW ORDER = 100001
  (2)  034276  001401                        BEQ     13$
  (3)  034300  104000                        EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
  (1)  034302                        13$:
  (1)  034302  005215                        INC     (R5)
  (1)
  (1)
12004                                ;*****************************************************************************
  (1)                                ;TEST:240       MUL       77777 * #77777 = 1 1        PS = 1
  (1)                                ;*****************************************************************************
  (1)
  (1)  034304                        TST240:
  (1)  034304  012703  077777                MOV     #77777,%3       ;LOAD MULTIPLICAND WITH 77777
  (1)  034310  070327  077777                MUL     #77777,%3       ;MULTIPLY 77777 * #77777
  (1)  034314  106737  025626                MFPS    @#PSWORD        ;SAVE PS
  (1)  034320  122737  000001  025626        CMPB    #1,@#PSWORD     ;IS PS = 1
  (2)  034326  001401                        BEQ     11$
  (3)  034330  104000                        EMT                     ;PS IS WRONG
  (1)  034332  022703  000001        11$:    CMP     #1,%3           ;IS HIGH ORDER = 1
  (2)  034336  001401                        BEQ     12$
  (3)  034340  104000                        EMT                     ;HIGH ORDER IS WRONG
  (1)  034342  022703  000001        12$:    CMP     #1,%3!1         ;IS LOW ORDER = 1
  (2)  034346  001401                        BEQ     13$
  (3)  034350  104000                        EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
  (1)  034352                        13$:
  (1)  034352  005215                        INC     (R5)
  (1)
  (1)
12005                                ;*****************************************************************************
  (1)                                ;TEST:241       MUL       2 * #2 = 4 4        PS = 0
  (1)                                ;*****************************************************************************
  (1)
  (1)  034354                        TST241:
  (1)  034354  010501                        MOV     R5,R1           ;SAVE R5
  (1)  034356  012705  000002                MOV     #2,%5           ;LOAD MULTIPLICAND WITH 2
  (1)  034362  070527  000002                MUL     #2,%5           ;MULTIPLY 2 * #2
  (1)  034366  106737  025626                MFPS    @#PSWORD        ;SAVE PS
  (1)  034372  122737  000000  025626        CMPB    #0,@#PSWORD     ;IS PS = 0
  (2)  034400  001401                        BEQ     11$
  (3)  034402  104000                        EMT                     ;PS IS WRONG
  (1)  034404  022705  000004        11$:    CMP     #4,%5           ;IS HIGH ORDER = 4
  (2)  034410  001401                        BEQ     12$
  (3)  034412  104000                        EMT                     ;HIGH ORDER IS WRONG
  (1)  034414  022705  000004        12$:    CMP     #4,%5!1         ;IS LOW ORDER = 4
  (2)  034420  001401                        BEQ     13$
  (3)  034422  104000                        EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
  (1)  034424                        13$:
  (1)  034424  010105                        MOV     R1,R5           ;RESTORE R5
```

K13



```
  (1)  034426  005215                          INC     (R5)
  (1)
  (1)
12006  034430  012702  040000                  MOV     #40000,%2
12007  034434  012703  025654                  MOV     #S5,%3
12008  034440  012704  025656                  MOV     #S6,%4
12009
12010                                  ;*****************************************************************
  (1)                                  ;TEST:242       MUL      125252 * S5 = 165252 100000        PS = 11
  (1)                                  ;*****************************************************************
  (1)
  (1)  034444                          TST242:
  (1)  034444  012700  125252                  MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
  (1)  034450  070067  171200                  MUL     S5,%0           ;MULTIPLY 125252 * S5
  (1)  034454  106737  025626                  MFPS    @#PSWORD        ;SAVE PS
  (1)  034460  122737  000011  025626          CMPB    #11,@#PSWORD    ;IS PS = 11
  (2)  034466  001401                          BEQ     11$
  (3)  034470  104000                          EMT                     ;PS IS WRONG
  (1)  034472  022700  165252          11$:    CMP     #165252,%0      ;IS HIGH ORDER = 165252
  (2)  034476  001401                          BEQ     12$
  (3)  034500  104000                          EMT                     ;HIGH ORDER IS WRONG
  (1)  034502  022701  100000          12$:    CMP     #100000,%0!1    ;IS LOW ORDER = 100000
  (2)  034506  001401                          BEQ     13$
  (3)  034510  104000                          EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
  (1)  034512                          13$:
  (1)  034512  005215                          INC     (R5)
  (1)
  (1)
12011                                  ;*****************************************************************
  (1)                                  ;TEST:243       MUL      125252 * @S6 = 165252 100000        PS = 11
  (1)                                  ;*****************************************************************
  (1)
  (1)  034514                          TST243:
  (1)  034514  012700  125252                  MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
  (1)  034520  070077  171132                  MUL     @S6,%0          ;MULTIPLY 125252 * @S6
  (1)  034524  106737  025626                  MFPS    @#PSWORD        ;SAVE PS
  (1)  034530  122737  000011  025626          CMPB    #11,@#PSWORD    ;IS PS = 11
  (2)  034536  001401                          BEQ     11$
  (3)  034540  104000                          EMT                     ;PS IS WRONG
  (1)  034542  022700  165252          11$:    CMP     #165252,%0      ;IS HIGH ORDER = 165252
  (2)  034546  001401                          BEQ     12$
  (3)  034550  104000                          EMT                     ;HIGH ORDER IS WRONG
  (1)  034552  022701  100000          12$:    CMP     #100000,%0!1    ;IS LOW ORDER = 100000
  (2)  034556  001401                          BEQ     13$
  (3)  034560  104000                          EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
  (1)  034562                          13$:
  (1)  034562  005215                          INC     (R5)
  (1)
  (1)
12012                                  ;*****************************************************************
  (1)                                  ;TEST:244       MUL      125252 * @#S5 = 165252 100000        PS = 11
  (1)                                  ;*****************************************************************
  (1)
  (1)  034564                          TST244:
  (1)  034564  012700  125252                  MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
  (1)  034570  070037  025654                  MUL     @#S5,%0         ;MULTIPLY 125252 * @#S5
```

```
  (1)  034574  106737  025626                    MFPS    @#PSWORD        ;SAVE PS
  (1)  034600  122737  000011  025626            CMPB    #11,@#PSWORD    ;IS PS = 11
  (2)  034606  001401                            BEQ     11$
  (3)  034610  104000                            EMT                     ;PS IS WRONG
  (1)  034612  022700  165252            11$:    CMP     #165252,#0      ;IS HIGH ORDER = 165252
  (2)  034616  001401                            BEQ     12$
  (3)  034620  104000                            EMT                     ;HIGH ORDER IS WRONG
  (1)  034622  022701  100000            12$:    CMP     #100000,#0!1    ;IS LOW ORDER = 100000
  (2)  034626  001401                            BEQ     13$
  (3)  034630  104000                            EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
  (1)  034632                            13$:
  (1)  034632  005215                            INC     (R5)
  (1)
  (1)
12013                                   ;*************************************************************************
  (1)                                   ;TEST:245       MUL     125252 * #2 = 165252 100000        PS = 11
  (1)                                   ;*************************************************************************
  (1)
  (1)  034634                           TST245:
  (1)  034634  012700  125252                    MOV     #125252,#0      ;LOAD MULTIPLICAND WITH 125252
  (1)  034640  070002                            MUL     #2,#0           ;MULTIPLY 125252 * #2
  (1)  034642  106737  025626                    MFPS    @#PSWORD        ;SAVE PS
  (1)  034646  122737  000011  025626            CMPB    #11,@#PSWORD    ;IS PS = 11
  (2)  034654  001401                            BEQ     11$
  (3)  034656  104000                            EMT                     ;PS IS WRONG
  (1)  034660  022700  165252            11$:    CMP     #165252,#0      ;IS HIGH ORDER = 165252
  (2)  034664  001401                            BEQ     12$
  (3)  034666  104000                            EMT                     ;HIGH ORDER IS WRONG
  (1)  034670  022701  100000            12$:    CMP     #100000,#0!1    ;IS LOW ORDER = 100000
  (2)  034674  001401                            BEQ     13$
  (3)  034676  104000                            EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
  (1)  034700                            13$:
  (1)  034700  005215                            INC     (R5)
  (1)
  (1)
12014                                   ;*************************************************************************
  (1)                                   ;TEST:246       MUL     125252 * (3)+ = 165252 100000      PS = 11
  (1)                                   ;*************************************************************************
  (1)
  (1)  034702                           TST246:
  (1)  034702  012700  125252                    MOV     #125252,#0      ;LOAD MULTIPLICAND WITH 125252
  (1)  034706  070023                            MUL     (3)+,#0         ;MULTIPLY 125252 * (3)+
  (1)  034710  106737  025626                    MFPS    @#PSWORD        ;SAVE PS
  (1)  034714  122737  000011  025626            CMPB    #11,@#PSWORD    ;IS PS = 11
  (2)  034722  001401                            BEQ     11$
  (3)  034724  104000                            EMT                     ;PS IS WRONG
  (1)  034726  022700  165252            11$:    CMP     #165252,#0      ;IS HIGH ORDER = 165252
  (2)  034732  001401                            BEQ     12$
  (3)  034734  104000                            EMT                     ;HIGH ORDER IS WRONG
  (1)  034736  022701  100000            12$:    CMP     #100000,#0!1    ;IS LOW ORDER = 100000
  (2)  034742  001401                            BEQ     13$
  (3)  034744  104000                            EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
  (1)  034746                            13$:
  (1)  034746  005215                            INC     (R5)
  (1)
  (1)
```

```
12015                                   ;*****************************************************************************
  (1)                                   ;TEST:247       MUL     125252 * -(3) = 165252 100000        PS = 11
  (1)                                   ;*****************************************************************************
  (1)
  (1)  034750                           TST247:
  (1)  034750  012700  125252                   MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
  (1)  034754  070043                           MUL     -(3),%0         ;MULTIPLY 125252 * -(3)
  (1)  034756  106737  025626                   MFPS    @#PSWORD        ;SAVE PS
  (1)  034762  122737  000011  025626           CMPB    #11,@#PSWORD    ;IS PS = 11
  (2)  034770  001401                           BEQ     11$
  (3)  034772  104000                           EMT                     ;PS IS WRONG
  (1)  034774  022700  165252           11$:    CMP     #165252,%0      ;IS HIGH ORDER = 165252
  (2)  035000  001401                           BEQ     12$
  (3)  035002  104000                           EMT                     ;HIGH ORDER IS WRONG
  (1)  035004  022701  100000           12$:    CMP     #100000,%0!1    ;IS LOW ORDER = 100000
  (2)  035010  001401                           BEQ     13$
  (3)  035012  104000                           EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
  (1)  035014                           13$:
  (1)  035014  005215                           INC     (R5)
  (1)
  (1)
12016                                   ;*****************************************************************************
  (1)                                   ;TEST:250       MUL     125252 * 2(4) = 165252 100000        PS = 11
  (1)                                   ;*****************************************************************************
  (1)
  (1)  035016                           TST250:
  (1)  035016  012700  125252                   MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
  (1)  035022  070064  000002                   MUL     2(4),%0         ;MULTIPLY 125252 * 2(4)
  (1)  035026  106737  025626                   MFPS    @#PSWORD        ;SAVE PS
  (1)  035032  122737  000011  025626           CMPB    #11,@#PSWORD    ;IS PS = 11
  (2)  035040  001401                           BEQ     11$
  (3)  035042  104000                           EMT                     ;PS IS WRONG
  (1)  035044  022700  165252           11$:    CMP     #165252,%0      ;IS HIGH ORDER = 165252
  (2)  035050  001401                           BEQ     12$
  (3)  035052  104000                           EMT                     ;HIGH ORDER IS WRONG
  (1)  035054  022701  100000           12$:    CMP     #100000,%0!1    ;IS LOW ORDER = 100000
  (2)  035060  001401                           BEQ     13$
  (3)  035062  104000                           EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
  (1)  035064                           13$:
  (1)  035064  005215                           INC     (R5)
  (1)
  (1)
12017                                   ;*****************************************************************************
  (1)                                   ;TEST:251       MUL     125252 * @(4) = 165252 100000        PS = 11
  (1)                                   ;*****************************************************************************
  (1)
  (1)  035066                           TST251:
  (1)  035066  012700  125252                   MOV     #125252,%0      ;LOAD MULTIPLICAND WITH 125252
  (1)  035072  070074  000000                   MUL     @(4),%0         ;MULTIPLY 125252 * @(4)
  (1)  035076  106737  025626                   MFPS    @#PSWORD        ;SAVE PS
  (1)  035102  122737  000011  025626           CMPB    #11,@#PSWORD    ;IS PS = 11
  (2)  035110  001401                           BEQ     11$
  (3)  035112  104000                           EMT                     ;PS IS WRONG
  (1)  035114  022700  165252           11$:    CMP     #165252,%0      ;IS HIGH ORDER = 165252
  (2)  035120  001401                           BEQ     12$
  (3)  035122  104000                           EMT                     ;HIGH ORDER IS WRONG
```

```
 (1)  035124  022701  100000          12$:    CMP     #100000,@#0!1   ;IS LOW ORDER = 100000
 (2)  035130  001401                          BEQ     13$
 (3)  035132  104000                          EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
 (1)  035134                          13$:
 (1)  035134  005215                          INC     (R5)
 (1)
 (1)
2018                                 ;*************************************************************************
 (1)                                 ;TEST:252       MUL     125252 * @(4)+ = 165252 100000          PS = 11
 (1)                                 ;*************************************************************************
 (1)
 (1)  035136                         TST252:
 (1)  035136  012700  125252                  MOV     #125252,@#0     ;LOAD MULTIPLICAND WITH 125252
 (1)  035142  070034                          MUL     @(4)+,@#0       ;MULTIPLY 125252 * @(4)+
 (1)  035144  106737  025626                  MFPS    @#PSWORD        ;SAVE PS
 (1)  035150  122737  000011  025626          CMPB    #11,@#PSWORD    ;IS PS = 11
 (2)  035156  001401                          BEQ     11$
 (3)  035160  104000                          EMT                     ;PS IS WRONG
 (1)  035162  022700  165252          11$:    CMP     #165252,@#0     ;IS HIGH ORDER = 165252
 (2)  035166  001401                          BEQ     12$
 (3)  035170  104000                          EMT                     ;HIGH ORDER IS WRONG
 (1)  035172  022701  100000          12$:    CMP     #100000,@#0!1   ;IS LOW ORDER = 100000
 (2)  035176  001401                          BEQ     13$
 (3)  035200  104000                          EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
 (1)  035202                          13$:
 (1)  035202  005215                          INC     (R5)
 (1)
 (1)
2019                                 ;*************************************************************************
 (1)                                 ;TEST:253       MUL     125252 * @-(4) = 165252 100000          PS = 11
 (1)                                 ;*************************************************************************
 (1)
 (1)  035204                         TST253:
 (1)  035204  012700  125252                  MOV     #125252,@#0     ;LOAD MULTIPLICAND WITH 125252
 (1)  035210  070054                          MUL     @-(4),@#0       ;MULTIPLY 125252 * @-(4)
 (1)  035212  106737  025626                  MFPS    @#PSWORD        ;SAVE PS
 (1)  035216  122737  000011  025626          CMPB    #11,@#PSWORD    ;IS PS = 11
 (2)  035224  001401                          BEQ     11$
 (3)  035226  104000                          EMT                     ;PS IS WRONG
 (1)  035230  022700  165252          11$:    CMP     #165252,@#0     ;IS HIGH ORDER = 165252
 (2)  035234  001401                          BEQ     12$
 (3)  035236  104000                          EMT                     ;HIGH ORDER IS WRONG
 (1)  035240  022701  100000          12$:    CMP     #100000,@#0!1   ;IS LOW ORDER = 100000
 (2)  035244  001401                          BEQ     13$
 (3)  035246  104000                          EMT                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
 (1)  035250                          13$:
 (1)  035250  005215                          INC     (R5)
 (1)
 (1)
```

```
12024                                   ;*********************************
12025                                   ;     DIV INSTRUCTION TESTS
12026                                   ;*********************************
12027
12028
12029
12030                                   ;*****************************************************************************
  (1)                                   ;TEST:254      DIV     0 4 / #2 = 2       REM = 0      PS = 0
  (1)                                   ;*****************************************************************************
  (1)
  (1)  035252                           TST254:
  (1)  035252  012700  000000                   MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
  (1)  035256  012701  000004                   MOV     #4,%0+1         ;LOAD LOW ORDER WITH 4
  (1)  035262  071027  000002                   DIV     #2,%0           ;DIVIDE BY #2
  (1)  035266  106737  025626                   MFPS    @#PSWORD        ;SAVE PS
  (1)  035272  122737  000000  025626           CMPB    #0,@#PSWORD     ;IS PS = 0
  (2)  035300  001401                           BEQ     11$
  (3)  035302  104000                           EMT                     ;PS IS WRONG
  (1)  035304  022700  000002           11$:    CMP     #2,%0           ;IS QUOTIENT = 2
  (2)  035310  001401                           BEQ     12$
  (3)  035312  104000                           EMT                     ;QUOTIENT IS WRONG
  (1)  035314  022701  000000           12$:    CMP     #0,%0+1         ;IS REMAINDER = 0
  (2)  035320  001401                           BEQ     13$
  (3)  035322  104000                           EMT                     ;WRONG REMAINDER
  (1)  035324                           13$:
  (1)  035324  005215                           INC     (R5)
  (1)
12031                                   ;*****************************************************************************
  (1)                                   ;TEST:255      DIV     -1 -9. / #3 = -3       REM = 0      PS = 10
  (1)                                   ;*****************************************************************************
  (1)
  (1)  035326                           TST255:
  (1)  035326  012702  177777                   MOV     #-1,%2          ;LOAD HIGH ORDER WITH -1
  (1)  035332  012703  177767                   MOV     #-9.,%2+1       ;LOAD LOW ORDER WITH -9.
  (1)  035336  071227  000003                   DIV     #3,%2           ;DIVIDE BY #3
  (1)  035342  106737  025626                   MFPS    @#PSWORD        ;SAVE PS
  (1)  035346  122737  000010  025626           CMPB    #10,@#PSWORD    ;IS PS = 10
  (2)  035354  001401                           BEQ     11$
  (3)  035356  104000                           EMT                     ;PS IS WRONG
  (1)  035360  022702  177775           11$:    CMP     #-3,%2          ;IS QUOTIENT = -3
  (2)  035364  001401                           BEQ     12$
  (3)  035366  104000                           EMT                     ;QUOTIENT IS WRONG
  (1)  035370  022703  000000           12$:    CMP     #0,%2+1         ;IS REMAINDER = 0
  (2)  035374  001401                           BEQ     13$
  (3)  035376  104000                           EMT                     ;WRONG REMAINDER
  (1)  035400                           13$:
  (1)  035400  005215                           INC     (R5)
  (1)
12032                                   ;*****************************************************************************
  (1)                                   ;TEST:256      DIV     0 9. / #2 = 4       REM = 1      PS = 0
  (1)                                   ;*****************************************************************************
  (1)
  (1)  035402                           TST256:
  (1)  035402  010501                           MOV     R5,R1           ;SAVE R5
  (1)  035404  012704  000000                   MOV     #0,%4           ;LOAD HIGH ORDER WITH 0
  (1)  035410  012705  000011                   MOV     #9.,%4+1        ;LOAD LOW ORDER WITH 9.
```

```
   (1)  035414  071427  000002                DIV     #2,%4           ;DIVIDE BY #2
   (1)  035420  106737  025626                MFPS    @#PSWORD        ;SAVE PS
   (1)  035424  122737  000000  025626        CMPB    #0,@#PSWORD     ;IS PS = 0
   (2)  035432  001401                        BEQ     11$
   (3)  035434  104000                        EMT                     ;PS IS WRONG
   (1)  035436  022704  000004          11$:  CMP     #4,%4           ;IS QUOTIENT = 4
   (2)  035442  001401                        BEQ     12$
   (3)  035444  104000                        EMT                     ;QUOTIENT IS WRONG
   (1)  035446  022705  000001          12$:  CMP     #1,%4+1         ;IS REMAINDER = 1
   (2)  035452  001401                        BEQ     13$
   (3)  035454  104000                        EMT                     ;WRONG REMAINDER
   (1)  035456                         13$:
   (1)  035456  010105                        MOV     R1,R5           ;RESTORE R5
   (1)  035460  005215                        INC     (R5)
   (1)
12033                                ;*************************************************************************
   (1)                               ;TEST:257      DIV     -1 -9. / #2 = -4       REM = -1      PS = 10
   (1)                               ;*************************************************************************
   (1)
   (1)  035462                       TST257:
   (1)  035462  012700  177777                MOV     #-1,%0          ;LOAD MIGH ORDER WITH -1
   (1)  035466  012701  177767                MOV     #-9.,%0+1       ;LOAD LOW ORDER WITH -9.
   (1)  035472  071027  000002                DIV     #2,%0           ;DIVIDE BY #2
   (1)  035476  106737  025626                MFPS    @#PSWORD        ;SAVE PS
   (1)  035502  122737  000010  025626        CMPB    #10,@#PSWORD    ;IS PS = 10
   (2)  035510  001401                        BEQ     11$
   (3)  035512  104000                        EMT                     ;PS IS WRONG
   (1)  035514  022700  177774          11$:  CMP     #-4,%0          ;IS QUOTIENT = -4
   (2)  035520  001401                        BEQ     12$
   (3)  035522  104000                        EMT                     ;QUOTIENT IS WRONG
   (1)  035524  022701  177777          12$:  CMP     #-1,%0+1        ;IS REMAINDER = -1
   (2)  035530  001401                        BEQ     13$
   (3)  035532  104000                        EMT                     ;WRONG REMAINDER
   (1)  035534                         13$:
   (1)  035534  005215                        INC     (R5)
   (1)
12034                                ;*************************************************************************
   (1)                               ;TEST:260      DIV     0 2 / #-3 = 0       REM = 2      PS = 4
   (1)                               ;*************************************************************************
   (1)
   (1)  035536                       TST260:
   (1)  035536  012702  000000                MOV     #0,%2           ;LOAD HIGH ORDER WITH 0
   (1)  035542  012703  000002                MOV     #2,%2+1         ;LOAD LOW ORDER WITH 2
   (1)  035546  071227  177775                DIV     #-3,%2          ;DIVIDE BY #-3
   (1)  035552  106737  025626                MFPS    @#PSWORD        ;SAVE PS
   (1)  035556  122737  000004  025626        CMPB    #4,@#PSWORD     ;IS PS = 4
   (2)  035564  001401                        BEQ     11$
   (3)  035566  104000                        EMT                     ;PS IS WRONG
   (1)  035570  022702  000000          11$:  CMP     #0,%2           ;IS QUOTIENT = 0
   (2)  035574  001401                        BEQ     12$
   (3)  035576  104000                        EMT                     ;QUOTIENT IS WRONG
   (1)  035600  022703  000002          12$:  CMP     #2,%2+1         ;IS REMAINDER = 2
   (2)  035604  001401                        BEQ     13$
   (3)  035606  104000                        EMT                     ;WRONG REMAINDER
   (1)  035610                         13$:
   (1)  035610  005215                        INC     (R5)
```

```
  (1)
12035                                   ;*****************************************************************
  (1)                                   ;TEST:261      DIV      -1 -2 / #3 = 0       REM = -2      PS = 4
  (1)                                   ;*****************************************************************
  (1)
  (1)  035612                           TST261:
  (1)  035612  010501                           MOV     R5,R1           ;SAVE R5
  (1)  035614  012704  177777                   MOV     #-1,%4          ;LOAD HIGH ORDER WITH -1
  (1)  035620  012705  177776                   MOV     #-2,%4+1        ;LOAD LOW ORDER WITH -2
  (1)  035624  071427  000003                   DIV     #3,%4           ;DIVIDE BY #3
  (1)  035630  106737  025626                   MFPS    @#PSWORD        ;SAVE PS
  (1)  035634  122737  000004  025626           CMPB    #4,@#PSWORD     ;IS PS = 4
  (2)  035642  001401                           BEQ     11$
  (3)  035644  104000                           EMT                     ;PS IS WRONG
  (1)  035646  022704  000000           11$:    CMP     #0,%4           ;IS QUOTIENT = 0
  (2)  035652  001401                           BEQ     12$
  (3)  035654  104000                           EMT                     ;QUOTIENT IS WRONG
  (1)  035656  022705  177776           12$:    CMP     #-2,%4+1        ;IS REMAINDER = -2
  (2)  035662  001401                           BEQ     13$
  (3)  035664  104000                           EMT                     ;WRONG REMAINDER
  (1)  035666                           13$:
  (1)  035666  010105                           MOV     R1,R5           ;RESTORE R5
  (1)  035670  005215                           INC     (R5)
  (1)
12036                                   ;*****************************************************************
  (1)                                   ;TEST:262      DIV      -1 -1 / #1 = -1       REM = 0      PS = 10
  (1)                                   ;*****************************************************************
  (1)
  (1)  035672                           TST262:
  (1)  035672  012700  177777                   MOV     #-1,%0          ;LOAD HIGH ORDER WITH -1
  (1)  035676  012701  177777                   MOV     #-1,%0+1        ;LOAD LOW ORDER WITH -1
  (1)  035702  071027  000001                   DIV     #1,%0           ;DIVIDE BY #1
  (1)  035706  106737  025626                   MFPS    @#PSWORD        ;SAVE PS
  (1)  035712  122737  000010  025626           CMPB    #10,@#PSWORD    ;IS PS = 10
  (2)  035720  001401                           BEQ     11$
  (3)  035722  104000                           EMT                     ;PS IS WRONG
  (1)  035724  022700  177777           11$:    CMP     #-1,%0          ;IS QUOTIENT = -1
  (2)  035730  001401                           BEQ     12$
  (3)  035732  104000                           EMT                     ;QUOTIENT IS WRONG
  (1)  035734  022701  000000           12$:    CMP     #0,%0+1         ;IS REMAINDER = 0
  (2)  035740  001401                           BEQ     13$
  (3)  035742  104000                           EMT                     ;WRONG REMAINDER
  (1)  035744                           13$:
  (1)  035744  005215                           INC     (R5)
  (1)
12037                                   ;*****************************************************************
  (1)                                   ;TEST:263      DIV      0 0 / #1 = 0       REM = 0      PS = 4
  (1)                                   ;*****************************************************************
  (1)
  (1)  035746                           TST263:
  (1)  035746  012700  000000                   MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
  (1)  035752  012701  000000                   MOV     #0,%0+1         ;LOAD LOW ORDER WITH 0
  (1)  035756  071027  000001                   DIV     #1,%0           ;DIVIDE BY #1
  (1)  035762  106737  025626                   MFPS    @#PSWORD        ;SAVE PS
  (1)  035766  122737  000004  025626           CMPB    #4,@#PSWORD     ;IS PS = 4
  (2)  035774  001401                           BEQ     11$
```

```
   (3)  035776  104000                          EMT                     ;PS IS WRONG
   (1)  036000  022700  000000          11$:    CMP     #0,#0           ;IS QUOTIENT = 0
   (2)  036004  001401                          BEQ     12$
   (3)  036006  104000                          EMT                     ;QUOTIENT IS WRONG
   (1)  036010  022701  000000          12$:    CMP     #0,#0+1         ;IS REMAINDER = 0
   (2)  036014  001401                          BEQ     13$
   (3)  036016  104000                          EMT                     ;WRONG REMAINDER
   (1)  036020                          13$:
   (1)  036020  005215                          INC     (R5)
   (1)
12038                                           ;*************************************************************************
   (1)                                          ;TEST:264      DIV     -1 125252 / #2 = 152525       REM = 0      PS = 10
   (1)                                          ;*************************************************************************
   (1)
   (1)  036022                          TST264:
   (1)  036022  012702  177777                  MOV     #-1,R2          ;LOAD HIGH ORDER WITH -1
   (1)  036026  012703  125252                  MOV     #125252,R2+1    ;LOAD LOW ORDER WITH 125252
   (1)  036032  071227  000002                  DIV     #2,R2           ;DIVIDE BY #2
   (1)  036036  106737  025626                  MFPS    @#PSWORD        ;SAVE PS
   (1)  036042  122737  000010  025626          CMPB    #10,@#PSWORD    ;IS PS = 10
   (2)  036050  001401                          BEQ     11$
   (3)  036052  104000                          EMT                     ;PS IS WRONG
   (1)  036054  022702  152525          11$:    CMP     #152525,R2      ;IS QUOTIENT = 152525
   (2)  036060  001401                          BEQ     12$
   (3)  036062  104000                          EMT                     ;QUOTIENT IS WRONG
   (1)  036064  022703  000000          12$:    CMP     #0,R2+1         ;IS REMAINDER = 0
   (2)  036070  001401                          BEQ     13$
   (3)  036072  104000                          EMT                     ;WRONG REMAINDER
   (1)  036074                          13$:
   (1)  036074  005215                          INC     (R5)
   (1)
12039                                           ;*************************************************************************
   (1)                                          ;TEST:265      DIV     -1 -1 / #-1 = 1       REM = 0      PS = 0
   (1)                                          ;*************************************************************************
   (1)
   (1)  036076                          TST265:
   (1)  036076  010501                          MOV     R5,R1           ;SAVE R5
   (1)  036100  012704  177777                  MOV     #-1,R4          ;LOAD HIGH ORDER WITH -1
   (1)  036104  012705  177777                  MOV     #-1,R4+1        ;LOAD LOW ORDER WITH -1
   (1)  036110  071427  177777                  DIV     #-1,R4          ;DIVIDE BY #-1
   (1)  036114  106737  025626                  MFPS    @#PSWORD        ;SAVE PS
   (1)  036120  122737  000000  025626          CMPB    #0,@#PSWORD     ;IS PS = 0
   (2)  036126  001401                          BEQ     11$
   (3)  036130  104000                          EMT                     ;PS IS WRONG
   (1)  036132  022704  000001          11$:    CMP     #1,R4           ;IS QUOTIENT = 1
   (2)  036136  001401                          BEQ     12$
   (3)  036140  104000                          EMT                     ;QUOTIENT IS WRONG
   (1)  036142  022705  000000          12$:    CMP     #0,R4+1         ;IS REMAINDER = 0
   (2)  036146  001401                          BEQ     13$
   (3)  036150  104000                          EMT                     ;WRONG REMAINDER
   (1)  036152                          13$:
   (1)  036152  010105                          MOV     R1,R5           ;RESTORE R5
   (1)  036154  005215                          INC     (R5)
   (1)
12040                                           ;*************************************************************************
   (1)                                          ;TEST:266      DIV     25253 1 / #125252 = 100000       REM = 1      PS = 10
```

```
  (1)                                   ;*****************************************************************
  (1)
  (1)  036156                           TST266:
  (1)  036156  012700  025253                   MOV     #25253,%0       ;LOAD HIGH ORDER WITH 25253
  (1)  036162  012701  000001                   MOV     #1,%0+1         ;LOAD LOW ORDER WITH 1
  (1)  036166  071027  125252                   DIV     #125252,%0      ;DIVIDE BY #125252
  (1)  036172  106737  025626                   MFPS    @#PSWORD        ;SAVE PS
  (1)  036176  122737  000010  025626           CMPB    #10,@#PSWORD    ;IS PS = 10
  (2)  036204  001401                           BEQ     11$
  (3)  036206  104000                           EMT                     ;PS IS WRONG
  (1)  036210  022700  100000           11$:    CMP     #100000,%0      ;IS QUOTIENT = 100000
  (2)  036214  001401                           BEQ     12$
  (3)  036216  104000                           EMT                     ;QUOTIENT IS WRONG
  (1)  036220  022701  000001           12$:    CMP     #1,%0+1         ;IS REMAINDER = 1
  (2)  036224  001401                           BEQ     13$
  (3)  036226  104000                           EMT                     ;WRONG REMAINDER
  (1)  036230                           13$:
  (1)  036230  005215                           INC     (R5)
  (1)
12041                                   ;*****************************************************************
  (1)                                   ;TEST:267      DIV     37777 77777 / #77777 = 77777       REM = 77776      PS = 0
  (1)                                   ;*****************************************************************
  (1)
  (1)  036232                           TST267:
  (1)  036232  012702  037777                   MOV     #37777,%2       ;LOAD HIGH ORDER WITH 37777
  (1)  036236  012703  077777                   MOV     #77777,%2+1     ;LOAD LOW ORDER WITH 77777
  (1)  036242  071227  077777                   DIV     #77777,%2       ;DIVIDE BY #77777
  (1)  036246  106737  025626                   MFPS    @#PSWORD        ;SAVE PS
  (1)  036252  122737  000000  025626           CMPB    #0,@#PSWORD     ;IS PS = 0
  (2)  036260  001401                           BEQ     11$
  (3)  036262  104000                           EMT                     ;PS IS WRONG
  (1)  036264  022702  077777           11$:    CMP     #77777,%2       ;IS QUOTIENT = 77777
  (2)  036270  001401                           BEQ     12$
  (3)  036272  104000                           EMT                     ;QUOTIENT IS WRONG
  (1)  036274  022703  077776           12$:    CMP     #77776,%2+1     ;IS REMAINDER = 77776
  (2)  036300  001401                           BEQ     13$
  (3)  036302  104000                           EMT                     ;WRONG REMAINDER
  (1)  036304                           13$:
  (1)  036304  005215                           INC     (R5)
  (1)
12042                                   ;*****************************************************************
  (1)                                   ;TEST:270      DIV     0 100000 / #2 = 40000       REM = 0      PS = 0
  (1)                                   ;*****************************************************************
  (1)
  (1)  036306                           TST270:
  (1)  036306  010501                           MOV     R5,R1           ;SAVE R5
  (1)  036310  012704  000000                   MOV     #0,%4           ;LOAD HIGH ORDER WITH 0
  (1)  036314  012705  100000                   MOV     #100000,%4+1    ;LOAD LOW ORDER WITH 100000
  (1)  036320  071427  000002                   DIV     #2,%4           ;DIVIDE BY #2
  (1)  036324  106737  025626                   MFPS    @#PSWORD        ;SAVE PS
  (1)  036330  122737  000000  025626           CMPB    #0,@#PSWORD     ;IS PS = 0
  (2)  036336  001401                           BEQ     11$
  (3)  036340  104000                           EMT                     ;PS IS WRONG
  (1)  036342  022704  040000           11$:    CMP     #40000,%4       ;IS QUOTIENT = 40000
  (2)  036346  001401                           BEQ     12$
  (3)  036350  104000                           EMT                     ;QUOTIENT IS WRONG
```

```
  (1)  036352  022705  000000          12$:    CMP     #0,%4+1          ;IS REMAINDER = 0
  (2)  036356  001401                          BEQ     13$
  (3)  036360  104000                          EMT                      ;WRONG REMAINDER
  (1)  036362                          13$:
  (1)  036362  010105                          MOV     R1,R5            ;RESTORE R5
  (1)  036364  005215                          INC     (R5)
  (1)
12043                                  ;*****************************************************************
  (1)                                  ;TEST:271      DIV     177777 77777 / #177776 = 40000       REM = 177777      PS = 0
  (1)                                  ;*****************************************************************
  (1)
  (1)  036366                          TST271:
  (1)  036366  012700  177777                  MOV     #177777,%0       ;LOAD HIGH ORDER WITH 177777
  (1)  036372  012701  077777                  MOV     #77777,%0+1      ;LOAD LOW ORDER WITH 77777
  (1)  036376  071027  177776                  DIV     #177776,%0       ;DIVIDE BY #177776
  (1)  036402  106737  025626                  MFPS    @#PSWORD         ;SAVE PS
  (1)  036406  122737  000000  025626          CMPB    #0,@#PSWORD      ;IS PS = 0
  (2)  036414  001401                          BEQ     11$
  (3)  036416  104000                          EMT                      ;PS IS WRONG
  (1)  036420  022700  040000          11$:    CMP     #40000,%0        ;IS QUOTIENT = 40000
  (2)  036424  001401                          BEQ     12$
  (3)  036426  104000                          EMT                      ;QUOTIENT IS WRONG
  (1)  036430  022701  177777          12$:    CMP     #177777,%0+1     ;IS REMAINDER = 177777
  (2)  036434  001401                          BEQ     13$
  (3)  036436  104000                          EMT                      ;WRONG REMAINDER
  (1)  036440                          13$:
  (1)  036440  005215                          INC     (R5)
  (1)
12044                                  ;*****************************************************************
  (1)                                  ;TEST:272      DIV     0 52525 / #52525 = 1       REM = 0      PS = 0
  (1)                                  ;*****************************************************************
  (1)
  (1)  036442                          TST272:
  (1)  036442  012702  000000                  MOV     #0,%2            ;LOAD HIGH ORDER WITH 0
  (1)  036446  012703  052525                  MOV     #52525,%2+1      ;LOAD LOW ORDER WITH 52525
  (1)  036452  071227  052525                  DIV     #52525,%2        ;DIVIDE BY #52525
  (1)  036456  106737  025626                  MFPS    @#PSWORD         ;SAVE PS
  (1)  036462  122737  000000  025626          CMPB    #0,@#PSWORD      ;IS PS = 0
  (2)  036470  001401                          BEQ     11$
  (3)  036472  104000                          EMT                      ;PS IS WRONG
  (1)  036474  022702  000001          11$:    CMP     #1,%2            ;IS QUOTIENT = 1
  (2)  036500  001401                          BEQ     12$
  (3)  036502  104000                          EMT                      ;QUOTIENT IS WRONG
  (1)  036504  022703  000000          12$:    CMP     #0,%2+1          ;IS REMAINDER = 0
  (2)  036510  001401                          BEQ     13$
  (3)  036512  104000                          EMT                      ;WRONG REMAINDER
  (1)  036514                          13$:
  (1)  036514  005215                          INC     (R5)
  (1)
12045                                  ;*****************************************************************
  (1)                                  ;TEST:273      DIV     0 77777 / #0 = DUMMY       REM = DUMMY      PS = 3
  (1)                                  ;*****************************************************************
  (1)
  (1)  036516                          TST273:
  (1)  036516  010501                          MOV     R5,R1            ;SAVE R5
  (1)  036520  012704  000000                  MOV     #0,%4            ;LOAD HIGH ORDER WITH 0
```

```
  (1) 036524 012705 077777           MOV     #77777,%4+1     ;LOAD LOW ORDER WITH 77777
  (1) 036530 071427 000000           DIV     #0,%4           ;DIVIDE BY #0
  (1) 036534 106737 025626           MFPS    @#PSWORD        ;SAVE PS
  (1) 036540 042737 000014 025626    BIC     #14,@#PSWORD
  (1) 036546 122737 000003 025626    CMPB    #3,@#PSWORD     ;IS PS = 3
  (2) 036554 001401                  BEQ     13$
  (3) 036556 104000                  EMT                     ;PS IS WRONG
  (1) 036560                      13$:
  (1) 036560 010105                  MOV     R1,R5           ;RESTORE R5
  (1) 036562 005215                  INC     (R5)
  (1)
12046                                ;*************************************************************************
  (1)                                ;TEST:274      DIV     77777 177777 / #2 = DUMMY       REM = DUMMY     PS = 2
  (1)                                ;*************************************************************************
  (1)
  (1) 036564                      TST274:
  (1) 036564 012700 077777           MOV     #77777,%0       ;LOAD HIGH ORDER WITH 77777
  (1) 036570 012701 177777           MOV     #177777,%0+1    ;LOAD LOW ORDER WITH 177777
  (1) 036574 071027 000002           DIV     #2,%0           ;DIVIDE BY #2
  (1) 036600 106737 025626           MFPS    @#PSWORD        ;SAVE PS
  (1) 036604 042737 000014 025626    BIC     #14,@#PSWORD
  (1) 036612 122737 000002 025626    CMPB    #2,@#PSWORD     ;IS PS = 2
  (2) 036620 001401                  BEQ     13$
  (3) 036622 104000                  EMT                     ;PS IS WRONG
  (1) 036624                      13$:
  (1) 036624 005215                  INC     (R5)
  (1)
12047 036626 012702 000002           MOV     #2,%2
12048 036632 012703 025664           MOV     #S9,%3
12049 036636 012704 025666           MOV     #S10,%4
12050
12051                                ;*************************************************************************
  (1)                                ;TEST:275      DIV     0 52525 / S9 = 25252       REM = 1      PS = 0
  (1)                                ;*************************************************************************
  (1)
  (1) 036642                      TST275:
  (1) 036642 012700 000000           MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
  (1) 036646 012701 052525           MOV     #52525,%0+1     ;LOAD LOW ORDER WITH 52525
  (1) 036652 071067 167006           DIV     S9,%0           ;DIVIDE BY S9
  (1) 036656 106737 025626           MFPS    @#PSWORD        ;SAVE PS
  (1) 036662 122737 000000 025626    CMPB    #0,@#PSWORD     ;IS PS = 0
  (2) 036670 001401                  BEQ     11$
  (3) 036672 104000                  EMT                     ;PS IS WRONG
  (1) 036674 022700 025252        11$:  CMP  #25252,%0       ;IS QUOTIENT = 25252
  (2) 036700 001401                  BEQ     12$
  (3) 036702 104000                  EMT                     ;QUOTIENT IS WRONG
  (1) 036704 022701 000001        12$:  CMP  #1,%0+1         ;IS REMAINDER = 1
  (2) 036710 001401                  BEQ     13$
  (3) 036712 104000                  EMT                     ;WRONG REMAINDER
  (1) 036714                      13$:
  (1) 036714 005215                  INC     (R5)
  (1)
12052                                ;*************************************************************************
  (1)                                ;TEST:276      DIV     0 52525 / @S10 = 25252       REM = 1      PS = 0
  (1)                                ;*************************************************************************
  (1)
```

```
  (1)  036716                          TST276:
  (1)  036716  012700  000000                 MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
  (1)  036722  012701  052525                 MOV     #52525,%0+1     ;LOAD LOW ORDER WITH 52525
  (1)  036726  071077  166734                 DIV     @S10,%0         ;DIVIDE BY @S10
  (1)  036732  106737  025626                 MFPS    @#PSWORD        ;SAVE PS
  (1)  036736  122737  000000  025626         CMPB    #0,@#PSWORD     ;IS PS = 0
  (2)  036744  001401                         BEQ     11$
  (3)  036746  104000                         EMT                     ;PS IS WRONG
  (1)  036750  022700  025252         11$:    CMP     #25252,%0       ;IS QUOTIENT = 25252
  (2)  036754  001401                         BEQ     12$
  (3)  036756  104000                         EMT                     ;QUOTIENT IS WRONG
  (1)  036760  022701  000001         12$:    CMP     #1,%0+1         ;IS REMAINDER = 1
  (2)  036764  001401                         BEQ     13$
  (3)  036766  104000                         EMT                     ;WRONG REMAINDER
  (1)  036770                          13$:
  (1)  036770  005215                         INC     (R5)
  (1)
12053                                  ;*************************************************************************
  (1)                                  ;TEST:277      DIV     0 52525 / @#S9 = 25252       REM = 1      PS = 0
  (1)                                  ;*************************************************************************
  (1)
  (1)  036772                          TST277:
  (1)  036772  012700  000000                 MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
  (1)  036776  012701  052525                 MOV     #52525,%0+1     ;LOAD LOW ORDER WITH 52525
  (1)  037002  071037  025664                 DIV     @#S9,%0         ;DIVIDE BY @#S9
  (1)  037006  106737  025626                 MFPS    @#PSWORD        ;SAVE PS
  (1)  037012  122737  000000  025626         CMPB    #0,@#PSWORD     ;IS PS = 0
  (2)  037020  001401                         BEQ     11$
  (3)  037022  104000                         EMT                     ;PS IS WRONG
  (1)  037024  022700  025252         11$:    CMP     #25252,%0       ;IS QUOTIENT = 25252
  (2)  037030  001401                         BEQ     12$
  (3)  037032  104000                         EMT                     ;QUOTIENT IS WRONG
  (1)  037034  022701  000001         12$:    CMP     #1,%0+1         ;IS REMAINDER = 1
  (2)  037040  001401                         BEQ     13$
  (3)  037042  104000                         EMT                     ;WRONG REMAINDER
  (1)  037044                          13$:
  (1)  037044  005215                         INC     (R5)
  (1)
12054                                  ;*************************************************************************
  (1)                                  ;TEST:300      DIV     0 52525 / %2 = 25252       REM = 1      PS = 0
  (1)                                  ;*************************************************************************
  (1)
  (1)  037046                          TST300:
  (1)  037046  012700  000000                 MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
  (1)  037052  012701  052525                 MOV     #52525,%0+1     ;LOAD LOW ORDER WITH 52525
  (1)  037056  071002                         DIV     %2,%0           ;DIVIDE BY %2
  (1)  037060  106737  025626                 MFPS    @#PSWORD        ;SAVE PS
  (1)  037064  122737  000000  025626         CMPB    #0,@#PSWORD     ;IS PS = 0
  (2)  037072  001401                         BEQ     11$
  (3)  037074  104000                         EMT                     ;PS IS WRONG
  (1)  037076  022700  025252         11$:    CMP     #25252,%0       ;IS QUOTIENT = 25252
  (2)  037102  001401                         BEQ     12$
  (3)  037104  104000                         EMT                     ;QUOTIENT IS WRONG
  (1)  037106  022701  000001         12$:    CMP     #1,%0+1         ;IS REMAINDER = 1
  (2)  037112  001401                         BEQ     13$
  (3)  037114  104000                         EMT                     ;WRONG REMAINDER
```

```
  (1)  037116                          13$:
  (1)  037116  005215                          INC     (R5)
  (1)
12055                                  ;*****************************************************************
  (1)                                  ;TEST:301      DIV     0 52525 / (3)+ = 25252       REM = 1      PS = 0
  (1)                                  ;*****************************************************************
  (1)
  (1)  037120                          TST301:
  (1)  037120  012700  000000                  MOV     #0,%0            ;LOAD HIGH ORDER WITH 0
  (1)  037124  012701  052525                  MOV     #52525,%0+1      ;LOAD LOW ORDER WITH 52525
  (1)  037130  071023                          DIV     (3)+,%0          ;DIVIDE BY (3)+
  (1)  037132  106737  025626                  MFPS    @#PSWORD         ;SAVE PS
  (1)  037136  122737  000000  025626          CMPB    #0,@#PSWORD      ;IS PS = 0
  (2)  037144  001401                          BEQ     11$
  (3)  037146  104000                          EMT                      ;PS IS WRONG
  (1)  037150  022700  025252          11$:    CMP     #25252,%0        ;IS QUOTIENT = 25252
  (2)  037154  001401                          BEQ     12$
  (3)  037156  104000                          EMT                      ;QUOTIENT IS WRONG
  (1)  037160  022701  000001          12$:    CMP     #1,%0+1          ;IS REMAINDER = 1
  (2)  037164  001401                          BEQ     13$
  (3)  037166  104000                          EMT                      ;WRONG REMAINDER
  (1)  037170                          13$:
  (1)  037170  005215                          INC     (R5)
  (1)
12056                                  ;*****************************************************************
  (1)                                  ;TEST:302      DIV     0 52525 / -(3) = 25252       REM = 1      PS = 0
  (1)                                  ;*****************************************************************
  (1)
  (1)  037172                          TST302:
  (1)  037172  012700  000000                  MOV     #0,%0            ;LOAD HIGH ORDER WITH 0
  (1)  037176  012701  052525                  MOV     #52525,%0+1      ;LOAD LOW ORDER WITH 52525
  (1)  037202  071043                          DIV     -(3),%0          ;DIVIDE BY -(3)
  (1)  037204  106737  025626                  MFPS    @#PSWORD         ;SAVE PS
  (1)  037210  122737  000000  025626          CMPB    #0,@#PSWORD      ;IS PS = 0
  (2)  037216  001401                          BEQ     11$
  (3)  037220  104000                          EMT                      ;PS IS WRONG
  (1)  037222  022700  025252          11$:    CMP     #25252,%0        ;IS QUOTIENT = 25252
  (2)  037226  001401                          BEQ     12$
  (3)  037230  104000                          EMT                      ;QUOTIENT IS WRONG
  (1)  037232  022701  000001          12$:    CMP     #1,%0+1          ;IS REMAINDER = 1
  (2)  037236  001401                          BEQ     13$
  (3)  037240  104000                          EMT                      ;WRONG REMAINDER
  (1)  037242                          13$:
  (1)  037242  005215                          INC     (R5)
  (1)
12057                                  ;*****************************************************************
  (1)                                  ;TEST:303      DIV     0 52525 / 2(4) = 25252       REM = 1      PS = 0
  (1)                                  ;*****************************************************************
  (1)
  (1)  037244                          TST303:
  (1)  037244  012700  000000                  MOV     #0,%0            ;LOAD HIGH ORDER WITH 0
  (1)  037250  012701  052525                  MOV     #52525,%0+1      ;LOAD LOW ORDER WITH 52525
  (1)  037254  071064  000002                  DIV     2(4),%0          ;DIVIDE BY 2(4)
  (1)  037260  106737  025626                  MFPS    @#PSWORD         ;SAVE PS
  (1)  037264  122737  000000  025626          CMPB    #0,@#PSWORD      ;IS PS = 0
  (2)  037272  001401                          BEQ     11$
```

```
  (3)  037274  104000                        EMT                         ;PS IS WRONG
  (1)  037276  022700  025252          11$:   CMP     #25252,%0           ;IS QUOTIENT = 25252
  (2)  037302  001401                        BEQ     12$
  (3)  037304  104000                        EMT                         ;QUOTIENT IS WRONG
  (1)  037306  022701  000001          12$:   CMP     #1,%0+1             ;IS REMAINDER = 1
  (2)  037312  001401                        BEQ     13$
  (3)  037314  104000                        EMT                         ;WRONG REMAINDER
  (1)  037316                         13$:
  (1)  037316  005215                        INC     (R5)
  (1)
12058                                       ;*************************************************************************
  (1)                                       ;TEST:304      DIV     0 52525 / @(4) = 25252       REM = 1      PS = 0
  (1)                                       ;*************************************************************************
  (1)
  (1)  037320                               TST304:
  (1)  037320  012700  000000                        MOV     #0,%0               ;LOAD HIGH ORDER WITH 0
  (1)  037324  012701  052525                        MOV     #52525,%0+1         ;LOAD LOW ORDER WITH 52525
  (1)  037330  071074  000000                        DIV     @(4),%0             ;DIVIDE BY @(4)
  (1)  037334  106737  025626                        MFPS    @#PSWORD            ;SAVE PS
  (1)  037340  122737  000000  025626                CMPB    #0,@#PSWORD         ;IS PS = 0
  (2)  037346  001401                                BEQ     11$
  (3)  037350  104000                                EMT                         ;PS IS WRONG
  (1)  037352  022700  025252          11$:          CMP     #25252,%0           ;IS QUOTIENT = 25252
  (2)  037356  001401                                BEQ     12$
  (3)  037360  104000                                EMT                         ;QUOTIENT IS WRONG
  (1)  037362  022701  000001          12$:          CMP     #1,%0+1             ;IS REMAINDER = 1
  (2)  037366  001401                                BEQ     13$
  (3)  037370  104000                                EMT                         ;WRONG REMAINDER
  (1)  037372                         13$:
  (1)  037372  005215                                INC     (R5)
  (1)
12059                                       ;*************************************************************************
  (1)                                       ;TEST:305      DIV     0 52525 / @(4)+ = 25252      REM = 1      PS = 0
  (1)                                       ;*************************************************************************
  (1)
  (1)  037374                               TST305:
  (1)  037374  012700  000000                        MOV     #0,%0               ;LOAD HIGH ORDER WITH 0
  (1)  037400  012701  052525                        MOV     #52525,%0+1         ;LOAD LOW ORDER WITH 52525
  (1)  037404  071034                                DIV     @(4)+,%0            ;DIVIDE BY @(4)+
  (1)  037406  106737  025626                        MFPS    @#PSWORD            ;SAVE PS
  (1)  037412  122737  000000  025626                CMPB    #0,@#PSWORD         ;IS PS = 0
  (2)  037420  001401                                BEQ     11$
  (3)  037422  104000                                EMT                         ;PS IS WRONG
  (1)  037424  022700  025252          11$:          CMP     #25252,%0           ;IS QUOTIENT = 25252
  (2)  037430  001401                                BEQ     12$
  (3)  037432  104000                                EMT                         ;QUOTIENT IS WRONG
  (1)  037434  022701  000001          12$:          CMP     #1,%0+1             ;IS REMAINDER = 1
  (2)  037440  001401                                BEQ     13$
  (3)  037442  104000                                EMT                         ;WRONG REMAINDER
  (1)  037444                         13$:
  (1)  037444  005215                                INC     (R5)
  (1)
12060                                       ;*************************************************************************
  (1)                                       ;TEST:306      DIV     0 52525 / @-(4) = 25252      REM = 1      PS = 0
  (1)                                       ;*************************************************************************
  (1)
```

```
  (1)  037446                          TST306:
  (1)  037446  012700  000000                 MOV     #0,%0           ;LOAD HIGH ORDER WITH 0
  (1)  037452  012701  052525                 MOV     #52525,%0+1     ;LOAD LOW ORDER WITH 52525
  (1)  037456  071054                         DIV     @-(4),%0        ;DIVIDE BY @-(4)
  (1)  037460  106737  025626                 MFPS    @#PSWORD        ;SAVE PS
  (1)  037464  122737  000000  025626         CMPB    #0,@#PSWORD     ;IS PS = 0
  (2)  037472  001401                         BEQ     11$
  (3)  037474  104000                         EMT                     ;PS IS WRONG
  (1)  037476  022700  025252         11$:    CMP     #25252,%0       ;IS QUOTIENT = 25252
  (2)  037502  001401                         BEQ     12$
  (3)  037504  104000                         EMT                     ;QUOTIENT IS WRONG
  (1)  037506  022701  000001         12$:    CMP     #1,%0+1         ;IS REMAINDER = 1
  (2)  037512  001401                         BEQ     13$
  (3)  037514  104000                         EMT                     ;WRONG REMAINDER
  (1)  037516                         13$:
  (1)  037516  005215                         INC     (R5)
  (1)
12061
12062
12063                                ;SPECIAL MULTIPLY DATA PATTERN TEST
12064
12065  037520  012701  177777         TSTSPC: MOV     #-1,R1          ;MAKE R1 -1 SO WE KNOW INSTR. WAS MODIFIER
12066  037524  012700  077700                 MOV     #77700,R0       ;SET UP TEST DATA
12067  037530  070027  000001                 MUL     #1,R0           ;DO MULTIPLY INSTRUCTION
12068  037534  022701  077700                 CMP     #77700,R1       ;CHECK LOW ORDER WORD
12069  037540  001401                         BEQ     1$
  (2)  037542  104000                         EMT                     ;LOW ORDER PRODUCT ERROR
12070  037544  005700                 1$:     TST     R0              ;CHECK HIGH ORDER WORD
12071  037546  001401                         BEQ     EISEND
  (2)  037550  104000                         EMT                     ;HIGH ORDER PRODUCT ERROR
12072  037552  000167  000026         EISEND: JMP     MMUTST          ;JMP OVER GARBAGE AND GET TO MMU TEST
```

```
12030
12081                           .SBTTL  MEMORY MANAGEMENT DEFINITIONS
  (1)
  (1)                           ;*KT11 VECTOR ADDRESS
  (1)
  (1)           000250          MMVEC=  250
  (1)
  (1)                           ;*KT11 STATUS REGISTER ADDRESSES
  (1)
  (1)           177572          SR0=    177572
  (1)           177574          SR1=    177574
  (1)           177576          SR2=    177576
  (1)           172516          SR3=    172516
  (1)
  (1)                           ;*USER "I" PAGE DESCRIPTOR REGISTERS
  (1)
  (1)           177600          UIPDR0= 177600
  (1)           177602          UIPDR1= 177602
  (1)           177604          UIPDR2= 177604
  (1)           177606          UIPDR3= 177606
  (1)           177610          UIPDR4= 177610
  (1)           177612          UIPDR5= 177612
  (1)           177614          UIPDR6= 177614
  (1)           177616          UIPDR7= 177616
  (1)
  (1)                           ;*USER "I" PAGE ADDRESS REGISTERS
  (1)
  (1)           177640          UIPAR0= 177640
  (1)           177642          UIPAR1= 177642
  (1)           177644          UIPAR2= 177644
  (1)           177646          UIPAR3= 177646
  (1)           177650          UIPAR4= 177650
  (1)           177652          UIPAR5= 177652
  (1)           177654          UIPAR6= 177654
  (1)           177656          UIPAR7= 177656
  (1)
  (1)                           ;*KERNEL "I" PAGE DESCRIPTOR REGISTERS
  (1)
  (1)           172300          KIPDR0= 172300
  (1)           172302          KIPDR1= 172302
  (1)           172304          KIPDR2= 172304
  (1)           172306          KIPDR3= 172306
  (1)           172310          KIPDR4= 172310
  (1)           172312          KIPDR5= 172312
  (1)           172314          KIPDR6= 172314
  (1)           172316          KIPDR7= 172316
  (1)
  (1)                           ;*KERNEL "I" PAGE ADDRESS REGISTERS
  (1)
  (1)           172340          KIPAR0= 172340
  (1)           172342          KIPAR1= 172342
  (1)           172344          KIPAR2= 172344
  (1)           172346          KIPAR3= 172346
  (1)           172350          KIPAR4= 172350
  (1)           172352          KIPAR5= 172352
  (1)           172354          KIPAR6= 172354
```

```
  (1)           172356                  KIPAR7= 172356
  (1)
12082           000006                  KSP=    SP
12083           000006                  USP=    SP
12084           177776                  PSW=    PS
12085           000020                  TBIT=   20
12086           000100                  WBIT=   100
12087           000001                  BIT0=   1
12088           000004                  ERRVEC= 4
12089
12090                                   ;*ADDITIONAL DEFINITIONS
12091                                   ;*
12092
12271
12272
  (1)  037556   000000                  WASR6:  .WORD   0        ;USED TO STORE THE STACK POINTER AFTER A TRAP
  (1)  037560   000000                  TRAPPC: .WORD   0        ;USED TO STORE THE PC OF A TRAP OR ABORT
  (1)  037562   000000                  TRAPPS: .WORD   0        ;USED TO STORE THE PS OF A TRAP OR ABORT
  (1)  037564   000000                  WASSR0: .WORD   0        ;USED TO STORE CONTENTS OF SR0
  (1)  037566   000000                  WASSR2: .WORD   0        ;USED TO STORE CONTENTS OR SR2
  (1)  037570   000000                  TBITPS: .WORD   0        ;SAVES THE PSW THAT MAY HAVE ITS T-BIT ON
  (1)  037572   000000                  $TMP0:  .WORD   0        ;TEMPORARY STORAGE LOCATION
  (1)  037574   000000                  $TMP1:  .WORD   0        ;TEMPORARY STORAGE LOCATION
  (1)  037576   000000                  $TMP2:  .WORD   0        ;TEMPORARY STORAGE LOCATION
  (1)  037600   000000                  $TMP3:  .WORD   0        ;TEMPORARY STORAGE LOCATION
  (1)  037602   000000                  $TMP4:  .WORD   0        ;TEMPORARY STORAGE LOCATION
12273
12274
```

```
12276
12277  037604  012706  001000          MMUTST: MOV     #STBOT,KSP      ;INITIALIZE THE STACK POINTER
12278  037610  012767  021550  140432          MOV     #T0250,MMVEC    ;LOAD MEMORY MANAGENT ROUTINE INTO VECTOR
12279  037616  012767  000340  140426          MOV     #340,MMVEC+2    ;SET NEW PS TO PRIORITY LEVEL 7-KERNEL
12280  037624  012767  000340  177736          MOV     #340,TBITPS     ;INITIALIZE LOG THAT HOLDS T-BIT PSW
12281  037632  005067  137734                  CLR     SR0             ;BE SURE MEM. MGMT IS OFF TO START WITH
12282  037636  000244                          CLZ                     ;CLR THE Z BIT
12283  037640  032777  000001  161730          BIT     #1,@SWR
12284  037646  001402                          BEQ     1$
12285  037650  000167  011102                  JMP     EXATST
12286  037654  012737  050646  000030  1$:     MOV     #ERROR3,@#30    ;SET UP EMT VECTOR TO GO TO RIGHT ERROR CALL
12287  037662  012737  000002  001004          MOV     #2,@#TESTN      ;INCREMENT TEST NUMBER
12288
```

```
                                        ;*******************************************************************************
                                        ;TEST 351       PSW PRIORITY BIT TEST
                                        ;*******************************************************************************
  037670                                TS351:
  037670  005000                                CLR     R0              ;INITIALIZE R0 WITH PRIORITY=0 DATA
  037672  005001                        2$:     CLR     R1              ;PREPARE R1 TO ACCEPT DATA READ
  037674  106400                                MTPS    R0              ;WRITE PRIORITY BITS IN THE PSW
  037676  106701                                MFPS    R1              ;READ BACK THE LOW BYTE OF PSW
  037700  042701  177437                        BIC     #177437,R1      ;MASK OFF EVERYTHING EXCEPT PRIORITY BITS
  037704  020001                                CMP     R0,R1           ;WAS CORRECT PRIORITY SET IN THE PSW?
  037706  001401                                BEQ     3$
  037710  104000                                EMT                     ;PRIORITY BITS SET WRONG IN PSW
                                                                        ;FOR TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
                                                                        ;"BR 2$" = 000770
  037712  062700  000040                3$:     ADD     #40,R0          ;CHANGE DATA TO NEXT PRIORITY
  037716  022700  000400                        CMP     #400,R0         ;HAVE PRIORITIES 0-7 ALL BEEN CHECKED?
  037722  001363                                BNE     2$              ;BRANCH IF NO

                                        ;*******************************************************************************
                                        ;TEST 352       PSW MODE BIT TEST
                                        ;*******************************************************************************
  037724                                TS352:
  037724  005000                                CLR     R0              ;INITIALIZE R0 WITH MODE BITS = 0000
  037726  005067  140044                2$:     CLR     PSW             ;INITIALIZE PSW
  037732  050067  140040                        BIS     R0,PSW          ;BIT SET THE PSW MODE BITS WITH R0
  037736  016701  140034                        MOV     PSW,R1          ;READ BACK THE CONTENTS OF THE PSW
  037742  042701  007777                        BIC     #007777,R1      ;MASK OFF EVERYTHING EXCEPT THE MODE BITS
  037746  020001                                CMP     R0,R1           ;WERE THE MODE BITS SET CORRECTLY?
  037750  001401                                BEQ     3$
  037752  104000                                EMT                     ;MODE BITS SET WRONG IN PSW
                                                                        ;FOR TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
                                                                        ;"BR 2$" = 000763
  037754  062700  010000                3$:     ADD     #10000,R0       ;CHANGE MODE BIT DATA
  037760  001362                                BNE     2$              ;BRANCH IF STILL MORE COMBINATIONS
  037762  005067  140010                        CLR     PSW             ;RESET PSW BEFORE LEAVING

                                        ;*******************************************************************************
                                        ;TEST 353       BYTE ADDRESSING TEST FOR PSW
                                        ;*******************************************************************************
  037766                                TS353:
  037766  005067  140004                2$:     CLR     PSW             ;CLEAR THE PSW
  037772  012700  000360                        MOV     #360,R0         ;PUT THE HIGH BYTE DATA INTO R0
  037776  110067  137775                        MOVB    R0,PSW+1        ;WRITE THE HIGH BYTE OF THE PSW
  040002  016701  137770                        MOV     PSW,R1          ;READ BACK THE ENTIRE PSW
  040006  042701  007437                        BIC     #007437,R1      ;MASK OFF THE T & CC BITS
  040012  000300                                SWAB    R0              ;GET DATA WRITTEN IN HIGH BYTE OF R0
  040014  020001                                CMP     R0,R1           ;WAS THE PSW WRITTEN TO CORRECTLY
  040016  001401                                BEQ     4$
  040020  104000                                EMT                     ;LOW BYTE EFFECTED BY WRITE TO HIGH BYTE OF PSW
                                                                        ;FOR TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
                                                                        ;"BR 2$" = 000760
```

```
12350  040022  005067  137750          4$:     CLR     PSW             ;CLEAR THE PSW
12351  040026  012700  000340                  MOV     #340,R0         ;PUT THE LOW BYTE DATA INTO R0
12352  040032  110067  137740                  MOVB    R0,PSW          ;WRITE THE LOW BYTE OF THE PSW
12353  040036  016701  137734                  MOV     PSW,R1          ;READ BACK THE ENTIRE PSW
12354  040042  042701  007437                  BIC     #007437,R1      ;MASK OFF THE T&CC BITS
12355  040046  020001                          CMP     R0,R1           ;WAS PSW WRITTEN TO CORRECTLY
12356  040050  001401                          BEQ     TS354
  (3)  040052  104000                          EMT                     ;HIGH BYTE EFFECTED BY WRITE TO LOW BYTE OF PSW
12357                                                                   ;FOR TIGHTER SCOPE LOOP
12358                                                                   ;REPLACE ERROR CALL WITH
12359                                                                   ;"BR 2$" = 000736
12360
12369                                  ;*****************************************************************************
  (2)                                  ;TEST 354       TEST AND SETUP OF STACK POINTERS
  (3)                                  ;*****************************************************************************
  (2)  040054                          TS354:
12370  040054  005067  137716                  CLR     PSW             ;GO TO KERNEL MODE
12371  040060  012706  001000                  MOV     #KERSTK,KSP     ;SET KERNEL STACK POINTER TO 1100
12372  040064  012767  140000  137704          MOV     #140000,PSW     ;GO TO USER MODE
12373  040072  012706  000600                  MOV     #USESTK,USP     ;SET USER STACK POINTER TO 700
12374  040076  005067  137674                  CLR     PSW             ;BACK TO KERNEL MODE
12375  040102  022706  001000                  CMP     #KERSTK,KSP     ;IS KERNEL R6 STILL 1100?
12376  040106  001401                          BEQ     TS355
  (3)  040110  104000                          EMT                     ;KERNEL R6 CHANGED BY WRITING USER R6
12377                                                                   ;FOR TIGHTER SCOPE LOOP
12378                                                                   ;REPLACE ERROR CALL WITH
12379                                                                   ;000756
12380
12381                                  ;*****************************************************************************
12382                                  ;*
12383                                  ;*      THE NEXT FIVE (5) TESTS WILL TRY TO ADDRESS ALL OF THE
12384                                  ;*      MEMORY MANAGEMENT REGISTERS (SR0,SR1,SR2,KERNEL & USER PAR/PDR'S).
12385                                  ;*      EVERY TIME A REGISTER TIMES OUT ITS ADDRESS WILL BE REPORTED.
12386                                  ;*      AT THE END OF EACH TEST A SUMMARY OF THE ADDRESSES THAT TIMED
12387                                  ;*      OUT DURING THAT TEST IS GIVEN.  THE RESULTS OF "AND-ING" AND "OR-ING"
12388                                  ;*      THEIR ADDRESSES IS GIVEN TO SHOW WHICH ADDRESS LINES MAY BE
12389                                  ;*      STUCK AT 0 OR 1.  THE PAR/PDR ADDRESS AND KT MUX'S ARE THE
12390                                  ;*      THINGS BEING CHECKED.
12391                                  ;*
12392                                  ;*****************************************************************************
12393
12403
12404                                  ;*****************************************************************************
  (2)                                  ;TEST 355       SR0,SR1,SR2,SR3 TIMEOUT TEST
  (3)                                  ;*****************************************************************************
  (2)  040112                          TS355:
12405  040112  012700  177572                  MOV     #SR0,R0         ;LOAD R0 WITH ADDRESS OF FIRST REG.
12406  040116  012701  000003                  MOV     #3,R1           ;LOAD R1 WITH THE LOOP COUNT
12407  040122  005710                  2$:     TST     (R0)            ;TRY ADDRESSING A STATUS REGISTER
12408                                                                   ;IF IT TIMES OUT GO TO 5$
12409  040124  062700  000002          3$:     ADD     #2,R0           ;PUT NEXT ADDRESS IN R0
12410  040130  077104                          SOB     R1,2$           ;LOOP BACK TO 2$ UNTIL ALL TESTED
12411  040132  005737  172516                  TST     @#172516        ;CHECK SR3 FOR RESPONSE
12412
12416                                  ;*****************************************************************************
  (2)                                  ;TEST 356       KERNEL PAR'S TIMEOUT TEST
```

```
  (3)                                   ;*****************************************************************************
  (2)  040136                           TS356:
12417
  (1)  040136  012700  172340                   MOV     #KIPAR0,R0      ;LOAD R0 WITH ADDRESS OF FIRST REG.
  (1)  040142  012701  000010                   MOV     #10,R1          ;LOAD R1 WITH LOOP COUNT (8)
  (1)  040146  005710                   2$:     TST     (R0)            ;TRY ADDRESSING A KIPAR
  (1)                                                                   ;IF IT TIMES OUT, WILL GO TO 5$
  (1)  040150  062700  000002           3$:     ADD     #2,R0           ;PUT NEXT KIPAR ADDRESS IN R0
  (1)  040154  077104                           SOB     R1,2$           ;LOOP BACK TO 2$ UNTIL ALL TESTED
12418
12422                                   ;*****************************************************************************
  (2)                                   ;TEST 357       KERNEL PDR'S TIMEOUT TEST
  (3)                                   ;*****************************************************************************
  (2)  040156                           TS357:
12423
  (1)  040156  012700  172300                   MOV     #KIPDR0,R0      ;LOAD R0 WITH ADDRESS OF FIRST REG.
  (1)  040162  012701  000010                   MOV     #10,R1          ;LOAD R1 WITH LOOP COUNT (8)
  (1)  040166  005710                   2$:     TST     (R0)            ;TRY ADDRESSING A KIPDR
  (1)                                                                   ;IF IT TIMES OUT, WILL GO TO 5$
  (1)  040170  062700  000002           3$:     ADD     #2,R0           ;PUT NEXT KIPDR ADDRESS IN R0
  (1)  040174  077104                           SOB     R1,2$           ;LOOP BACK TO 2$ UNTIL ALL TESTED
12424
12428                                   ;*****************************************************************************
  (2)                                   ;TEST 360       USER PAR'S TIMEOUT TEST
  (3)                                   ;*****************************************************************************
  (2)  040176                           TS360:
12429
  (1)  040176  012700  177640                   MOV     #UIPAR0,R0      ;LOAD R0 WITH ADDRESS OF FIRST REG.
  (1)  040202  012701  000010                   MOV     #10,R1          ;LOAD R1 WITH LOOP COUNT (8)
  (1)  040206  005710                   2$:     TST     (R0)            ;TRY ADDRESSING A UIPAR
  (1)                                                                   ;IF IT TIMFS OUT, WILL GO TO 5$
  (1)  040210  062700  000002           3$:     ADD     #2,R0           ;PUT NEXT UIPAR ADDRESS IN R0
  (1)  040214  077104                           SOB     R1,2$           ;LOOP BACK TO 2$ UNTIL ALL TESTED
12430
12434                                   ;*****************************************************************************
  (2)                                   ;TEST 361       USER PDR'S TIMEOUT TEST
  (3)                                   ;*****************************************************************************
  (2)  040216                           TS361:
12435
  (1)  040216  012700  177600                   MOV     #UIPDR0,R0      ;LOAD R0 WITH ADDRESS OF FIRST REG.
  (1)  040222  012701  000010                   MOV     #10,R1          ;LOAD R1 WITH LOOP COUNT (8)
  (1)  040226  005710                   2$:     TST     (R0)            ;TRY ADDRESSING A UIPDR
  (1)                                                                   ;IF IT TIMES OUT, WILL GO TO 5$
  (1)  040230  062700  000002           3$:     ADD     #2,R0           ;PUT NEXT UIPDR ADDRESS IN R0
  (1)  040234  077104                           SOB     R1,2$           ;LOOP BACK TO 2$ UNTIL ALL TESTED
12436
12447                                   ;*****************************************************************************
  (2)                                   ;TEST 362       SR0(15:13) BIT TEST & SR2 TEST
  (3)                                   ;*****************************************************************************
  (2)  040236                           TS362:
12448
12449  040236  012700  177572           1$:     MOV     #SR0,R0         ;LOAD ADDRESS OF SR0 INTO R0
12450  040242  012710  160000                   MOV     #160000,(R0)    ;SET BITS <15:13> IN SR0 (ERROR BITS)
12451  040246  000005                           RESET                   ;ISSUE AND "INIT" SIGNAL
12452  040250  011001                           MOV     (R0),R1         ;READ SR0 INTO R1 TO SEE IF CLEAR
12453  040252  001401                           BEQ     2$
```

```
  (2)  040254  104000                          EMT                     ;SRO<15:13> NOT CLEARED BY A "RESET"
12454                                                                   ;FOR TIGHTER SCOPE LOOP
12455                                                                   ;REPLACE ERROR CALL WITH
12456                                                                   ;"BR 1$" = 000770
12457  040256  016767  137314  177302  2$:     MOV     SR2,WASSR2      ;READ CONTENTS OF SR2
12458  040264  012701  040256                  MOV     #2$,R1          ;LOAD EXPECTED CONTENTS INTO R1
12459  040270  020167  177272                  CMP     R1,WASSR2       ;IS SR2 TRACKING?
12460  040274  001401                          BEQ     3$
  (2)  040276  104000                          EMT                     ;SR2 NOT "TRACKING" VIRTUAL ADDRESSES
12461                                                                   ;FOR TIGHTER SCOPE LOOP
12462                                                                   ;REPLACE ERROR CALL WITH
12463                                                                   ;"BR 2$" = 000767
12464  040300  012701  100000          3$:     MOV     #100000,R1      ;PUT DATA TO BE WRITTEN IN R1
12465  040304  012703  000003                  MOV     #3,R3           ;SETUP R3 AS A LOOP COUNTER
12466  040310  005010                  4$:     CLR     (R0)            ;CLEAR SRO
12467  040312  050110                  5$:     BIS     R1,(R0)         ;SET ONE OF THE ERROR BITS IN SRO
12468  040314  011002                          MOV     (R0),R2         ;READ SRO INTO R2
12469  040316  020102                          CMP     R1,R2           ;DID RIGHT ERROR BIT GET SET?
12470  040320  001401                          BEQ     6$
  (2)  040322  104000                          EMT                     ;BITS WERE SET WRONG IN SRO
12471                                                                   ;FOR TIGHTER SCOPE LOOP
12472                                                                   ;REPLACE ERROR CALL WITH
12473                                                                   ;"BR 4$" = 000772
12474  040324  012704  040312          6$:     MOV     #5$,R4          ;LOAD EXPECTED CONTENTS OF SR2 IN R4
12475  040330  016767  137242  177230          MOV     SR2,WASSR2      ;READ SR2
12476  040336  020467  177224                  CMP     R4,WASSR2       ;DID SR2 LOCK UP WHEN ERROR
12477                                                                   ;BIT SET IN SR1?
12478  040342  001401                          BEQ     7$
  (2)  040344  104000                          EMT                     ;SR2 DID NOT LOCK UP
12479                                                                   ;FOR TIGHTER SCOPE LOOP
12480                                                                   ;REPLACE ERROR CALL WITH
12481                                                                   ;"BR 4$" = 000761
12482  040346  006001                  7$:     ROR     R1              ;CHANGE DATA TO CHECK NEXT ERROR BIT
12483  040350  077321                          SOB     R3,4$           ;LOOP BACK UNTIL <15:13> ALL TESTED
12484  040352  005010                          CLR     (R0)            ;CLEAR SRO BEFORE LEAVING
12485
12495                                  ;*****************************************************************************
  (2)                                  ;TEST 363       SRO & PSW DUAL ADDRESSING TEST
  (3)                                  ;*****************************************************************************
  (2)  040354                          TS363:
12496
12497  040354  005067  137416          1$:     CLR     PSW             ;CLEAR THE PSW
12498  040360  005067  137206                  CLR     SRO             ;CLEAR STATUS REGISTER 0
12499  040364  106427  000340                  MTPS    #340            ;SET PRIORITY 7 IN LOW BYTE OF PSW
12500  040370  016700  137176                  MOV     SRO,R0          ;READ STATUS REGISTER 0
12501  040374  001401                          BEQ     2$
  (2)  040376  104000                          EMT                     ;SRO EFFECTED BY A WRITE TO THE PSW
12502                                                                   ;FOR TIGHTER SCOPE LOOP
12503                                                                   ;REPLACE ERROR CALL WITH
12504                                                                   ;"BR 1$" = 000767
12505  040400  005067  137166          2$:     CLR     SRO             ;BE SURE SRO IS 0 BEFORE LEAVING
12506  040404  005067  137366                  CLR     PSW             ;BE SURE PSW IS 0 BEFORE LEAVING
12507
12515                                  ;*****************************************************************************
  (2)                                  ;TEST 364       TEST THAT SR1 READS ALL ZEROS
  (3)                                  ;*****************************************************************************
```

```
  (2) 040410                          TS364:
12516 040410 012700 177777            1$:   MOV   #-1,R0          ;FILL R0 WITH ALL ONES
12517 040414 016700 137154                  MOV   SR1,R0          ;READ SR1 INTO R0
12518 040420 001401                         BEQ   2$
  (2) 040422 104000                         EMT                   ;SR1 DID NOT READ ALL ZEROS
12519                                                             ;FOR TIGHTER SCOPE LOOP
12520                                                             ;REPLACE ERROR CALL WITH
12521                                                             ;000772
12522 040424 012767 177777 132064   2$:   MOV   #-1,SR3          ;TRY TO WRITE ONES TO SR3
12523 040432 022767 000060 132056         CMP   #60,SR3         ;ONLY BITS <5:4> SHOULD BE ONES
12524 040440 001401                       BEQ   3$
  (2) 040442 104000                       EMT                   ;DIDN'T READ BACK A "60"
12525 040444 004567 010152          3$:   JSR   R5,CHKAPT
12526 040450 000402                       BR    90$
12527 040452 000005                       RESET                 ;CLEARS SR3
12528 040454 000402                       BR    91$
12529 040456 005067 132034          90$:  CLR   SR3
12530 040462 005767 132030          91$:  TST   SR3             ;VERIFY THAT IT WAS CLEARED
12531 040466                        4$:
  (2) 040466 001401                       BEQ   TS365
  (3) 040470 104000                       EMT                   ;SR3 DIDN'T READ ALL ZEROS
12532
12540
12541                               ;NOTE   F11 CHANGES INCLUDED CHECKING ALL BITS<15:0> OF PARS
12542                               ;       INSTEAD OF ONLY BITS<11:0>.
12543
12544                               ;*****************************************************************************
  (2)                               ;TEST 365       BIT TEST OF KERNEL & USER PAR'S
  (3)                               ;*****************************************************************************
  (2) 040472                        TS365:
12545
12546 040472 012700 172340          1$:   MOV   #KIPAR0,R0      ;LOAD ADDRESS OF FIRST PAR IN R0
12547 040476 012703 000010          2$:   MOV   #10,R3          ;SETUP R3 TO COUNT 8 PAR'S
12548 040502 005010                 3$:   CLR   (R0)            ;CLEAR THE PAR
12549 040504 011001                       MOV   (R0),R1         ;READ THE PAR INTO R1
12550 040506 001401                       BEQ   4$
  (2) 040510 104000                       EMT                   ;PAR WOULD NOT CLEAR
12551                                                           ;FOR TIGHTER SCOPE LOOP
12552                                                           ;REPLACE ERROR CALL WITH
12553                                                           ;"BR 3$" = 000774
12554 040512 012704 077777          4$:   MOV   #077777,R4      ;LOAD "WALKING 0" TEST PATTERN IN R4
12555 040516 005010                 5$:   CLR   (R0)            ;CLEAR THE PAR BEFORE LOADING DATA
12556 040520 050410                       BIS   R4,(R0)         ;BIT SET THE TEST PATTERN INTO THE PAR
12557 040522 011002                       MOV   (R0),R2         ;READ THE PAR INTO R2
12558 040524 020402                       CMP   R4,R2           ;DOES DATA WRITTEN=DATA READ?
12559 040526 001401                       BEQ   6$
  (2) 040530 104000                       EMT                   ;PAR BITS DID NOT SET CORRECTLY
12560                                                           ;FOR TIGHTER SCOPE LOOP
12561                                                           ;REPLACE ERROR CALL WITH
12562                                                           ;"BR 5$" = 000767
12563 040532 000261                 6$:   SEC                   ;SET THE C-BIT FOR THE ROTATE INST.
12564 040534 006004                       ROR   R4              ;ROTATE THE TEST PATTERN IN R4
12565 040536 103767                       BCS   5$              ;BRANCH BACK IF MORE BITS TO TEST
12566 040540 062700 000002                ADD   #2,R0           ;GET NEXT PAR ADDRESS IN R0
12567 040544 077322                       SOB   R3,3$           ;BRANCH BACK UNTIL ALL PAR'S TESTED
12568 040546 022700 177660                CMP   #UIPAR7+2,R0    ;HAVE USER PAR'S BEEN TESTED
```

```
12569  040552  103003                  BHIS    TS366           ;GET TO NEXT TEST
12570  040554  012700  177640          MOV     #UIPAR0,R0      ;LOAD FIRST USER PAR ADDR. IN R0
12571  040560  000746                  BR      2$              ;BRANCH BACK TO TEST USER PAR'S
12572                                                          ;LEAVE TEST WITH BITS <11:1>=1 IN ALL PAR'S
12581                          ;*****************************************************************************
  (2)                          ;TEST 366       BIT TEST OF KERNEL & USER PDR'S
  (3)                          ;*****************************************************************************
  (2)  040562                  TS366:
12582
12583  040562  012700  172300  1$:     MOV     #KIPDR0,R0      ;LOAD ADDRESS OF FIRST PDR IN R0
12584  040566  012703  000010  2$:     MOV     #10,R3          ;SETUP R3 TO COUNT 8 PDR'S
12585  040572  005010          3$:     CLR     (R0)            ;CLEAR THE PDR
12586  040574  011001                  MOV     (R0),R1         ;READ THE PDR INTO R1
12587  040576  001401                  BEQ     4$
  (2)  040600  104000                  EMT                     ;PDR WOULD NOT CLEAR
12588                                                          ;FOR TIGHTER SCOPE LOOP
12589                                                          ;REPLACE ERROR CALL WITH
12590                                                          ;"BR 3$" = 000774
12591  040602  012704  077777  4$:     MOV     #077777,R4      ;LOAD "WALKING "0" TEST PATTERN IN R4
12592  040606  005010          5$:     CLR     (R0)            ;CLEAR THE PDR BEFORE LOADING DATA
12593  040610  010401                  MOV     R4,R1           ;LOAD DATA INTO R1
12594  040612  042701  100361          BIC     #100361,R1      ;MASK UNUSED BITS OUT OF THE DATA
12595  040616  050110                  BIS     R1,(R0)         ;BIT SET THE TEST PATTERN INTO THE PDR
12596  040620  011002                  MOV     (R0),R2         ;READ THE PDR INTO R2
12597  040622  020102                  CMP     R1,R2           ;DOES DATA WRITTEN=DATA READ?
12598  040624  001401                  BEQ     6$
  (2)  040626  104000                  EMT                     ;PDR BITS DID NOT SET CORRECTLY
12599                                                          ;FOR TIGHTER SCOPE LOOP
12600                                                          ;REPLACE ERROR CALL WITH
12601                                                          ;"BR 5$" = 000767
12602  040630  000261          6$:     SEC                     ;SET THE C-BIT FOR THE ROTATE INST.
12603  040632  006004                  ROR     R4              ;ROTATE THE TEST PATTERN IN R4
12604  040634  103764                  BCS     5$              ;BRANCH BACK IF MORE BITS TO TEST
12605  040636  062700  000002          ADD     #2,R0           ;GET NEXT PDR ADDRESS IN R0
12606  040642  077325                  SOB     R3,3$           ;BRANCH BACK UNTIL ALL PDR'S TESTED
12607  040644  022700  177620          CMP     #UIPDR7+2,R0    ;HAVE USER PDR'S BEEN TESTED?
12608  040650  103003                  BHIS    TS367           ;GET TO NEXT TEST
12609  040652  012700  177600          MOV     #UIPDR0,R0      ;LOAD FIRST USER PDR ADDR. IN R0
12610  040656  000743                  BR      2$              ;BRANCH BACK TO TEST USER PDR'S
12611                                                          ;LEAVE TEST WITH ALL WRITEABLE BITS IN
12612                                                          ;ALL PDR'S = 1
12613
12621                          ;*****************************************************************************
  (2)                          ;TEST 367       TEST FOR DUAL BYTE ADDRESSING OF KERNEL & USER PAR'S
  (3)                          ;*****************************************************************************
  (2)  040660                  TS367:
12622
12623  040660  012700  172340  1$:     MOV     #KIPAR0,R0      ;LOAD ADDRESS OF FIRST PAR INTO R0
12624  040664  012703  000010          MOV     #10,R3          ;LOAD LOOP COUNTER TO DO 8 PAR'S
12625  040670  012701  177777  3$:     MOV     #-1,R1          ;LOAD TEST PATTERN INTO R1
12626  040674  005010                  CLR     (R0)            ;CLEAR THE PAR
12627  040676  110110                  MOVB    R1,(R0)         ;WRITE 1'S TO THE LOW BYTE OF THE PAR
12628  040700  011002                  MOV     (R0),R2         ;READ THE ENTIRE PAR INTO R2
12629  040702  042701  177400          BIC     #177400,R1      ;MASK HIGH BYTE & UNUSED BITS OUT OF THE DATA
12630  040706  020102                  CMP     R1,R2           ;WAS ONLY THE LOW BYTE WRITTEN TO
12631  040710  001401                  BEQ     5$
```

```
  (2)  040712  104000                        EMT                     ;HIGH BYTE EFFECTED BY WRITING LOW BYTE IN PAR
12632                                                                 ;FOR TIGHTER SCOPE LOOP
12633                                                                 ;REPLACE ERROR CALL WITH
12634                                                                 ;"BR 3$" = 000766
12635  040714  005010                    5$:     CLR     (R0)        ;CLEAR THE PAR
12636  040716  012701  177777                    MOV     #-1,R1      ;LOAD TEST, PATTERN INTO R1
12637  040722  110160  000001                    MOVB    R1,1(R0)    ;WRITE 1'S TO THE HIGH BYTE OF THE PAR
12638  040726  011002                            MOV     (R0),R2     ;READ THE ENTIRE PAR INTO R2
12639                                                                 ;F11 CHANGE     WAS #170377
12640  040730  042701  000377                    BIC     #000377,R1  ;MASK LOW BYTE & UNUSED BITS OUT OF DATA
12641  040734  020102                            CMP     R1,R2       ;WAS ONLY THE HIGH BYTE WRITTEN TO?
12642  040736  001401                            BEQ     6$
  (2)  040740  104000                            EMT                 ;LOW BYTE EFFECTED BY WRITING HIGH BYTE IN PAR
12643                                                                 ;FOR TIGHTER SCOPE LOOP
12644                                                                 ;REPLACE ERROR CALL WITH
12645                                                                 ;"BR 5" = 000765
12646  040742  062700  000002            6$:     ADD     #2,R0       ;PUT ADDRESS OF NEXT PAR IN R0
12647  040746  077330                            SOB     R3,3$       ;BRANCH BACK UNTIL 8 PAR'S TESTED
12648  040750  022700  177660                    CMP     #UIPAR7+2,R0  ;HAVE USER PAR'S BEEN TESTED
12649  040754  103003                            BHIS    TS370       ;GET TO NEXT TEST
12650  040756  012700  177640                    MOV     #UIPAR0,R0  ;LOAD ADDRESS OF FIRST USER PAR IN R0
12651  040762  000742                            BR      3$          ;BRANCH BACK TO TEST USER PAR'S
12652
12660                                 ;*****************************************************************************
  (2)                                 ;TEST 370       TEST FOR DUAL BYTE ADDRESSING OF KERNEL & USER PDR'S
  (3)                                 ;*****************************************************************************
  (2)  040764                       TS370:
12661
12662  040764  012700  172300            1$:     MOV     #KIPDR0,R0  ;LOAD ADDRESS OF FIRST PDR INTO R0
12663  040770  012703  000010                    MOV     #10,R3      ;LOAD LOOP COUNTER TO DO 8 PDR'S
12664  040774  012701  177777            3$:     MOV     #-1,R1      ;LOAD TEST PATTERN INTO R1
12665  041000  005010                            CLR     (R0)        ;CLEAR THE PDR
12666  041002  110110                            MOVB    R1,(R0)     ;WRITE 1'S TO THE LOW BYTE OF THE PDR
12667  041004  011002                            MOV     (R0),R2     ;READ THE ENTIRE PDR INTO R2
12668  041006  042701  177761                    BIC     #177761,R1  ;MASK HIGH BYTE & UNUSED BITS OUT OF DATA
12669  041012  020102                            CMP     R1,R2       ;WAS ONLY THE LOW BYTE WRITTEN TO?
12670  041014  001401                            BEQ     5$
  (2)  041016  104000                            EMT                 ;HIGH BYTE EFFECTED BY WRITING LOW BYTE IN PDR
12671                                                                 ;FOR TIGHTER SCOPE LOOP
12672                                                                 ;REPLACE ERROR CALL WITH
12673                                                                 ;"BR 3$" = 000766
12674  041020  005010            5$:     CLR     (R0)                ;CLEAR THE PDR
12675  041022  012701  177777                    MOV     #-1,R1      ;LOAD TEST PATTERN INTO R1
12676  041026  110160  000001                    MOVB    R1,1(R0)    ;WRITE 1'S TO THE HIGH BYTE OF THE PDR
12677  041032  011002                            MOV     (R0),R2     ;READ THE ENTIRE PDR INTO R2
12678  041034  042701  100377                    BIC     #100377,R1  ;MASK LOW BYTE & UNUSED BITS OUT OF DATA
12679  041040  020102                            CMP     R1,R2       ;WAS ONLY THE HIGH BYTE WRITTEN TO?
12680  041042  001401                            BEQ     6$
  (2)  041044  104000                            EMT                 ;LOW BYTE EFFECTED BY WRITING HIGH BYTE IN PDR
12681                                                                 ;FOR TIGHTER SCOPE LOOP
12682                                                                 ;REPLACE ERROR CALL WITH
12683                                                                 ;"BR 5$" = 000765
12684  041046  062700  000002            6$:     ADD     #2,R0       ;PUT ADDRESS OF NEXT PDR IN R0
12685  041052  077330                            SOB     R3,3$       ;BRANCH BACK UNTIL 8 PDR'S TESTED
12686  041054  022700  177620                    CMP     #UIPDR7+2,R0  ;HAVE USER PDR'S BEEN TESTED?
12687  041060  103003                            BHIS    TS371       ;GET TO NEXT TEST
```

```
12688  041062  012700  177600                MOV     #UIPDR0,R0      ;LOAD ADDRESS OF FIRST USER PDR IN R0
12689  041066  000742                        BR      3$              ;BRANCH BACK TO TEST USER PDR'S
12690
12702                          ;*****************************************************************************
  (2)                          ;TEST 371       PAR-PDR DUAL ADDRESSING TEST
  (3)                          ;*****************************************************************************
  (2)  041070                  TS371:
12703
12704  041070  012703  000010                MOV     #10,R3          ;LOAD LOOP COUNTER WITH AN 8
12705  041074  012700  172300                MOV     #KIPDR0,R0      ;LOAD ADDRESS OF FIRST KERNEL PDR AND R0
12706  041100  004767  007242                JSR     PC,SETREG       ;SET ALL BITS IN ALL PAR'S IN PDR'S
12707  041104  012706  001000        2$:     MOV     #KERSTK,KSP     ;SETUP STACK POINTER
12708  041110  005010                        CLR     (R0)            ;CLEAR ONE OF THE KERNEL PDR'S
12709  041112  004767  007322                JSR     PC,CMPREG       ;SEE IF OTHER PAR/PDR'S WERE EFFECTED
12710  041116  012720  177777                MOV     #-1,(R0)+       ;RESTORE ALL ONES, AND SETUP FOR NEXT PDR
12711  041122  077310                        SOB     R3,2$           ;LOOP TO 2$ UNTIL ALL KERNEL PDR'S CHECKED
12712  041124  012703  000010                MOV     #10,R3          ;LOAD LOOP COUNTER WITH AN 8
12713  041130  012700  172340                MOV     #KIPAR0,R0      ;LOAD ADDRESS OF FIRST KERNEL PAR IN R0
12714  041134  012706  001000        3$:     MOV     #KERSTK,KSP     ;SETUP STACK POINTER
12715  041140  005010                        CLR     (R0)            ;CLEAR ONE OF THE KERNEL PAR'S
12716  041142  004767  007272                JSR     PC,CMPREG       ;SEE IF OTHER PAR/PDR'S WERE EFFECTED
12717  041146  012720  177777                MOV     #-1,(R0)+       ;RESTORE ALL ONES, AND SETUP FOR NEXT PAR
12718  041152  077310                        SOB     R3,3$           ;LOOP TO 3$ UNTIL ALL KERNEL PAR'S CHECKED
12719  041154  012703  000010                MOV     #10,R3          ;LOAD LOOP COUNTER WITH AN 8
12720  041160  012700  177600                MOV     #UIPDR0,R0      ;LOAD ADDRESS OF FIRST USER PDR IN R0
12721  041164  012706  001000        4$:     MOV     #KERSTK,KSP     ;SETUP STACK POINTER
12722  041170  005010                        CLR     (R0)            ;CLEAR ONE OF THE USER PDR'S
12723  041172  004767  007242                JSR     PC,CMPREG       ;SEE IF OTHER PAR/PDR'S WERE EFFECTED
12724  041176  012720  177777                MOV     #-1,(R0)+       ;RESTORE ALL ONES, AND SETUP FOR NEXT UPDR
12725  041202  077310                        SOB     R3,4$           ;LOOP TO 4$ UNTIL ALL USER PDR'S CHECKED
12726  041204  012703  000010                MOV     #10,R3          ;LOAD LOOP COUNTER WITH AN 8
12727  041210  012700  177640                MOV     #UIPAR0,R0      ;LOAD ADDRESS OF FIRST USER PAR IN R0
12728  041214  012706  001000        5$:     MOV     #KERSTK,KSP     ;SETUP STACK POINTER
12729  041220  005010                        CLR     (R0)            ;CLEAR ONE OF THE USER PAR'S
12730  041222  004767  007212                JSR     PC,CMPREG       ;SEE IF OTHER PAR/PDR'S WERE EFFECTED
12731  041226  012720  177777                MOV     #-1,(R0)+       ;RESTORE ALL ONES, AND SETUP FOR NEXT UPAR
12732  041232  077310                        SOB     R3,5$           ;LOOP TO 5$ UNTIL ALL USER PAR'S CHECKED
12733
12743                          ;*****************************************************************************
  (2)                          ;TEST 372       TEST THAT PAR-PDR'S NOT AFFECTED BY RESET
  (3)                          ;*****************************************************************************
  (2)  041234                  TS372:
12744
12745
12746  041234  032737  000001  001020        BIT     #1, @#$ENV      ;ARE WE RUNNING UNDER APT
  (1)  041242  001403                        BEQ     70$             ;IF NO THEN DO TEST
  (1)  041244  005737  001006                TST     @#$PASS         ;IS THIS FIRST PASS
  (2)  041250  001063                        BNE     TS373           ;IF NO THEN SHIP TO NEXT TEST
  (1)  041252                  70$:
12747  041252  004767  007070        1$:     JSR     PC,SETREG       ;SET ALL BITS IN ALL PAR'S AND PDR'S
12748  041256  000005                        RESET                   ;ISSUE AN "INIT" BY EXECUTING A RESET
12749  041260  012700  172300        10$:    MOV     #KIPDR0,R0      ;LOAD ADDRESS OF FIRST KERNEL PDR IN R0
12750  041264  012704  000010                MOV     #10,R4          ;LOAD LOOP COUNTER WITH AN 8
12751  041270  011001                2$:     MOV     (R0),R1         ;READ A KERNEL PDR INTO R1
12752  041272  022701  077416                CMP     #77416,R1       ;ARE ALL THE BITS STILL SET?
12753  041276  001401                        BEQ     3$
```

```
  (2) 041300  104000                      EMT                     ;KERNEL PDR AFFECTED BY A RESET
12754                                                             ;FOR TIGHTER SCOPE LOOP
12755                                                             ;REPLACE ERROR CALL WITH
12756                                                             ;"BR 2$" = 000773
12757 041302  062700  000002        3$:   ADD     #2,R0           ;FORM ADDRESS OF NEXT KERNEL PDR
12758 041306  077410                      SOB     R4,2$           ;LOOP TO 2$ UNTIL ALL KERNEL PDR'S CHECKED
12759 041310  012700  172340              MOV     #KIPAR0,R0      ;LOAD ADDRESS OF FIRST KERNEL PAR IN R0
12760 041314  012704  000010              MOV     #10,R4          ;LOAD LOOP COUNTER WITH AN 8
12761 041320  011001              4$:     MOV     (R0),R1         ;READ A KERNEL PAR INTO R1
12762                                                             ;****F11 CHANGE****     WAS #7777
12763 041322  022701  177777              CMP     #177777,R1      ;ARE ALL THE BITS STILL SET?
12764 041326  001401                      BEQ     5$
  (2) 041330  104000                      EMT                     ;KERNEL PAR AFFECTED BY A RESET
12765                                                             ;FOR TIGHTER SCOPE LOOP
12766                                                             ;REPLACE ERROR CALL WITH
12767                                                             ;"BR 4$" = 000773
12768 041332  062700  000002        5$:   ADD     #2,R0           ;FORM ADDRESS OF NEXT KERNEL PAR
12769 041336  077410                      SOB     R4,4$           ;LOOP TO 4$ UNTIL ALL KERNEL PAR'S CHECKED
12770 041340  012700  177600              MOV     #UIPDR0,R0      ;LOAD ADDRESS OF FIRST USER PDR IN R0
12771 041344  012704  000010              MOV     #10,R4          ;LOAD LOOP COUNTER WITH AN 8
12772 041350  011001              6$:     MOV     (R0),R1         ;READ A USER PDR INTO R1
12773 041352  022701  077416              CMP     #77416,R1       ;ARE ALL THE BITS STILL SET?
12774 041356  001401                      BEQ     7$
  (2) 041360  104000                      EMT                     ;USER PDR AFFECTED BY A RESET
12775                                                             ;FOR TIGHTER SCOPE LOOP
12776                                                             ;REPLACE ERROR CALL WITH
12777                                                             ;"BR 6$" = 000773
12778 041362  062700  000002        7$:   ADD     #2,R0           ;FORM ADDRESS OF NEXT USER PDR
12779 041366  077410                      SOB     R4,6$           ;LOOP TO 6$ UNTIL ALL USER PDR'S CHECKED
12780
12781 041370  012700  177640              MOV     #UIPAR0,R0      ;LOAD ADDRESS OF FIRST USER PAR IN R0
12782 041374  012704  000010              MOV     #10,R4          ;LOAD LOOP COUNTER WITH AN 8
12783 041400  011001              8$:     MOV     (R0),R1         ;READ A USER PAR INTO R1
12784                                                             ;****F11 CHANGE****     WAS #7777
12785 041402  022701  177777              CMP     #177777,R1      ;ARE ALL THE BITS STILL SET?
12786 041406  001401                      BEQ     9$
  (2) 041410  104000                      EMT                     ;USER PAR AFFECTED BY A RESET
12787                                                             ;FOR TIGHTER SCOPE LOOP
12788                                                             ;REPLACE ERROR CALL WITH
12789                                                             ;"BR 8$" = 000773
12790 041412  062700  000002        9$:   ADD     #2,R0           ;FORM ADDRESS OF NEXT USER PAR
12791 041416  077410                      SOB     R4,8$           ;LOOP TO 8$ UNTIL ALL USER PAR'S CHECKED
12792
12793
12794
12795
12796
12797
12798
12799
12800
12801
12802
12803
12804
12805
```

```
12806
12807
12808
12809
12810
12811
12812
12813
12814
12832                           ;*****************************************************************************************
  (2)                           ;TEST 373       RELOCATION & ADDER TEST (NO CARRIES)
  (3)                           ;*****************************************************************************************
  (2)  041420                   TS373:
12833
12834  041420  012700  172340   1$:     MOV     #KIPAR0,R0      ;LOAD ADDRESS OF FIRST KERNEL PAR IN R0
12835  041424  005001                   CLR     R1              ;CLEAR R1
12836  041426  012702  000007           MOV     #7,R2           ;LOAD LOOP COUNTER WITH A 7
12837  041432  010120           2$:     MOV     R1,(R0)+        ;MAP KERNEL PAR'S TO PAGES 0-6 (4K EACH)
12838  041434  062701  000200           ADD     #200,R1
12839  041440  077204                   SOB     R2,2$           ;LOOP UNTIL KIPAR0 - KIPAR6 ARE LOADED
12840  041442  012710  177600           MOV     #177600,(R0)    ;MAP KIPAR7 TO THE I/O PAGE
12841  041446  012700  172300           MOV     #KIPDR0,R0      ;LOAD ADDRESS OF FIRST KERNEL PDR IN R0
12842  041452  012701  077406           MOV     #77406,R1       ;LOAD PDR DATA INTO R1
12843  041456  012702  000010           MOV     #10,R2          ;LOAD LOOP COUNTER WITH AN 8
12844  041462  010120           3$:     MOV     R1,(R0)+        ;MAP ALL 8 PAGES 128 BLOCKS, UPWARD
12845  041464  077202                   SOB     R2,3$           ;       EXPANDABLE, READ/WRITE
12846
12847  041466  012700  067776   4$:     MOV     #67776,R0       ;LOAD PHYSICAL ADDR. PBA INTO R0
  (1)  041472  012701  107776           MOV     #107776,R1      ;LOAD VIRTUAL ADDR. VBA INTO R1
  (1)  041476  012702  125250           MOV     #125250,R2      ;LOAD TEST PATTERN INTO R2
  (1)  041502  012704  000600           MOV     #600,R4 ;LOAD R4 WITH PAR VALUE
  (1)  041506  010467  130636           MOV     R4,KIPAR4       ;LOAD KERNEL PAR 4 BITS <11:00>
  (1)  041512  011067  176054           MOV     (R0),$TMP0      ;SAVE CONTENTS AT TEST LOCATION
  (1)  041516  005067  130774           CLR     SR3             ;SET UP FOR 18-BIT ADDRESSING
  (1)  041522  052767  000001  136042   BIS     #BIT0,SR0       ;TURN ON "RELOCATION"
  (1)  041530  010211                   MOV     R2,(R1)         ;LOAD 125250 USING ADDER  (PAR4 + VIRT ADDR.)
  (1)  041532  005067  136034           CLR     SR0             ;TURN OFF MEMORY MGMT.
  (1)  041536  011003                   MOV     (R0),R3         ;READ 125250 BACK WITHOUT USING MEM. MGMT.
  (1)  041540  016710  176026           MOV     $TMP0,(R0)      ;RESTORE ORIGINAL CONTENTS TO TEST LOC.
  (1)  041544  020203                   CMP     R2,R3           ;WAS SAME PATTERN READ BACK THAT WAS
  (1)                                                           ;WRITTEN USING "DEST-ONLY-RELOC."?
  (2)  041546  001401                   BEQ     5$
  (3)  041550  104000                   EMT                     ;TEST LOCATION DID NOT HAVE PATTERN
  (1)                                                           ;THAT SHOULD HAVE BEEN WRITTEN TO IT.
  (1)                                                           ;APPARENTLY PHYSICAL ADDR. WAS
  (1)                                                           ;FORMED WRONG BY ADDERS USING
  (1)                                                           ;THE VIRTUAL ADDR. AND KIPAR4
  (1)                                                           ;FOR TIGHTER SCOPE LOOP
  (1)                                                           ;REPLACE ERROR CALL WITH
  (1)                                                           ;"BR 4$" = 000742
  (1)  041552                   5$:
12848  041552  012700  067776   6$:     MOV     #67776,R0       ;LOAD PHYSICAL ADDR. PBA INTO R0
  (1)  041556  012701  102576           MOV     #102576,R1      ;LOAD VIRTUAL ADDR. VBA INTO R1
  (1)  041562  012702  125251           MOV     #125251,R2      ;LOAD TEST PATTERN INTO R2
  (1)  041566  012704  000652           MOV     #652,R4 ;LOAD R4 WITH PAR VALUE
  (1)  041572  010467  130552           MOV     R4,KIPAR4       ;LOAD KERNEL PAR 4 BITS <11:00>
```

```
  (1)  041576  011067  175770              MOV     (R0),$TMP0      ;SAVE CONTENTS AT TEST LOCATION
  (1)  041602  005067  130710              CLR     SR3             ;SET UP FOR 18-BIT ADDRESSING
  (1)  041606  052767  000001  135756      BIS     #BIT0,SR0       ;TURN ON "RELOCATION"
  (1)  041614  010211                      MOV     R2,(R1)         ;LOAD 125251 USING ADDER  (PAR4 + VIRT ADDR.)
  (1)  041616  005067  135750              CLR     SR0             ;TURN OFF MEMORY MGMT.
  (1)  041622  011003                      MOV     (R0),R3         ;READ 125251 BACK WITHOUT USING MEM. MGMT.
  (1)  041624  016710  175742              MOV     $TMP0,(R0)      ;RESTORE ORIGINAL CONTENTS TO TEST LOC.
  (1)  041630  020203                      CMP     R2,R3           ;WAS SAME PATTERN READ BACK THAT WAS
  (1)                                                              ;WRITTEN USING "DEST-ONLY-RELOC."?
  (2)  041632  001401                      BEQ     7$
  (3)  041634  104000                      EMT                     ;TEST LOCATION DID NOT HAVE PATTERN
  (1)                                                              ;THAT SHOULD HAVE BEEN WRITTEN TO IT.
  (1)                                                              ;APPARENTLY PHYSICAL ADDR. WAS
  (1)                                                              ;FORMED WRONG BY ADDERS USING
  (1)                                                              ;THE VIRTUAL ADDR. AND KIPAR4
  (1)                                                              ;FOR TIGHTER SCOPE LOOP
  (1)                                                              ;REPLACE ERROR CALL WITH
  (1)                                                              ;"BR 6$" = 000742
  (1)  041636                      7$:
12849  041636  012700  067776      8$:     MOV     #67776,R0       ;LOAD PHYSICAL ADDR. PBA INTO R0
  (1)  041642  012701  105276              MOV     #105276,R1      ;LOAD VIRTUAL ADDR. VBA INTO R1
  (1)  041646  012702  125252              MOV     #125252,R2      ;LOAD TEST PATTERN INTO R2
  (1)  041652  012704  000625              MOV     #625,R4 ;LOAD R4 WITH PAR VALUE
  (1)  041656  010467  130466              MOV     R4,KIPAR4       ;LOAD KERNEL PAR 4 BITS <11:00>
  (1)  041662  011067  175704              MOV     (R0),$TMP0      ;SAVE CONTENTS AT TEST LOCATION
  (1)  041666  052767  000020  130622      BIS     #BIT4,SR3       ;SET UP FOR 22-BIT ADDRESSING
  (1)  041674  052767  000001  135670      BIS     #BIT0,SR0       ;TURN ON "RELOCATION"
  (1)  041702  010211                      MOV     R2,(R1)         ;LOAD 125252 USING ADDER  (PAR4 + VIRT ADDR.)
  (1)  041704  005067  135662              CLR     SR0             ;TURN OFF MEMORY MGMT.
  (1)  041710  011003                      MOV     (R0),R3         ;READ 125252 BACK WITHOUT USING MEM. MGMT.
  (1)  041712  016710  175654              MOV     $TMP0,(R0)      ;RESTORE ORIGINAL CONTENTS TO TEST LOC.
  (1)  041716  020203                      CMP     R2,R3           ;WAS SAME PATTERN READ BACK THAT WAS
  (1)                                                              ;WRITTEN USING "DEST-ONLY-RELOC."?
  (2)  041720  001401                      BEQ     9$
  (3)  041722  104000                      EMT                     ;TEST LOCATION DID NOT HAVE PATTERN
  (1)                                                              ;THAT SHOULD HAVE BEEN WRITTEN TO IT.
  (1)                                                              ;APPARENTLY PHYSICAL ADDR. WAS
  (1)                                                              ;FORMED WRONG BY ADDERS USING
  (1)                                                              ;THE VIRTUAL ADDR. AND KIPAR4
  (1)                                                              ;FOR TIGHTER SCOPE LOOP
  (1)                                                              ;REPLACE ERROR CALL WITH
  (1)                                                              ;"BR 8$" = 000742
  (1)  041724                      9$:
12850
12851  041724  012700  177776              MOV     @PSW,R0         ;LOAD PHYS. ADDR. OF PSW INTO R0
12852  041730  012701  100076              MOV     #100076,R1      ;LOAD VIRTUAL ADDR. FOR PSW INTO R1
12853  041734  012702  030340              MOV     #030340,R2      ;LOAD DATA FOR PSW IN R2
12854  041740  012704  007777              MOV     #7777,R4        ;LOAD R4 WITH PAR VALUE
12855  041744  010467  130400              MOV     R4,KIPAR4       ;LOAD KERNEL PAR 4 BITS <11:00>
12856  041750  005010                      CLR     (R0)            ;CLEAR THE PSW
12857  041752  005067  130540              CLR     SR3             ;SET UP FOR 18-BIT ADDRESSING
12858  041756  052767  000001  135606      BIS     #BIT0,SR0       ;TURN ON "MEMORY MANAGEMENT"
12859  041764  010211                      MOV     R2,(R1)         ;LOAD PSW USING ADDER (PAR4 + VIRT ADDR.)
12860  041766  005067  135600              CLR     SR0             ;TURN OFF MEM. MGMT (SR0=0)
12861  041772  011003                      MOV     (R0),R3         ;READ PSW BACK WITHOUT USING MEM. MGMT.
12862  041774  005010                      CLR     (R0)            ;CLEAR THE PSW
```

```
2864  041776  042703  000037          BIC     #37,R3          ;MASK T-BIT & CC BITS OUT OF DATA READ
2865  042002  020203                  CMP     R2,R3           ;WAS PSW WRITTEN?
2866  042004  001401                  BEQ     11$
 (2)  042006  104000                  EMT                     ;PSW DID NOT HAVE DATA THAT IT SHOULD HAVE.
2867                                                          ;APPARENTLY PHYS. ADDR. OF PSW WAS
2868                                                          ;NOT FORMED BY ADDERS USING THE
2869                                                          ;VIRTUAL ADDR. AND KIPAR4
2870                                                          ;FOR TIGHTER SCOPE LOOP
2871                                                          ;REPLACE ERROR CALL WITH
2872                                                          ;"BR 10$" = 000742
2873  042010  012700  177776      11$:    MOV     #PSW,R0         ;LOAD PHYS. ADDR. OF PSW INTO R0
2874  042014  012701  117776              MOV     #117776,R1      ;LOAD VIRTUAL ADDR. FOR PSW INTO R1
2875  042020  012702  030240              MOV     #030240,R2      ;LOAD DATA FOR PSW IN R2
2876  042024  012704  177600              MOV     #177600,R4      ;LOAD R4 WITH PAR VALUE
2877  042030  010467  130314              MOV     R4,KIPAR4       ;LOAD KERNEL PAR 4 BITS <11:00>
2878  042034  052767  000020  130454      BIS     #BIT4,SR3       ;SET UP FOR 22-BIT ADDRESSING
2879  042042  052767  000001  135522      BIS     #BIT0,SR0       ;TURN ON "MEMORY MANAGEMENT"
2880  042050  010211                      MOV     R2,(R1)         ;LOAD PSW USING ADDER (PAR4 + VIRT. ADDR.)
2881  042052  005067  135514              CLR     SR0             ;TURN OFF MEM. MGMT (SR0=0)
2882  042056  011003                      MOV     (R0),R3         ;READ PSW BACK WITHOUT USING MEM. MGMT.
2883  042060  005010                      CLR     (R0)            ;CLEAR THE PSW
2884  042062  042703  000037              BIC     #37,R3          ;MASK T-BIT & CC BITS OUT OF DATA READ
2885  042066  020203                      CMP     R2,R3           ;WAS PSW WRITTEN WHILE IN MAINT. MODE?
2886  042070  001401                      BEQ     TS374
 (3)  042072  104000                      EMT                     ;PSW DID NOT HAVE DATA THAT IT SHOULD
2887                                                          ;HAVE, APPARENTLY PHYS. ADDR. OF PSW WAS
2888                                                          ;NOT FORMED BY ADDERS USING THE
2889                                                          ;VIRTUAL ADDR. AND KIPAR4
2890                                                          ;FOR TIGHTER SCOPE LOOP
2891                                                          ;REPLACE ERROR CALL WITH
2892                                                          ;"BR 11$" = 000743
2893
2905                                  ;*****************************************************************************
 (2)                                  ;TEST 374       RELOCATION & ADDER TEST (WITH CARRIES)
 (3)                                  ;*****************************************************************************
 (2)  042074                          TS374:
2906
2907  042074                          1$:                     ;KERNEL PAR'S AND PDR'S HAVE BEEN
2908                                                          ;SETUP BY THE PREVIOUS TEST
2909  042074  012700  066476          2$:     MOV     #66476,R0       ;LOAD PHYSICAL ADDR. PBA INTO R0
 (1)  042100  012701  114376              MOV     #114376,R1      ;LOAD VIRTUAL ADDR. VBA INTO R1
 (1)  042104  012702  125253              MOV     #125253,R2      ;LOAD TEST PATTERN INTO R2
 (1)  042110  012704  000521              MOV     #521,R4 ;LOAD R4 WITH PAR VALUE
 (1)  042114  010467  130230              MOV     R4,KIPAR4       ;LOAD KERNEL PAR 4 BITS <11:00>
 (1)  042120  011067  175446              MOV     (R0),$TMP0      ;SAVE CONTENTS AT TEST LOCATION
 (1)  042124  052767  000020  130364      BIS     #BIT4,SR3       ;SET UP FOR 22-BIT ADDRESSING
 (1)  042132  052767  000001  135432      BIS     #BIT0,SR0       ;TURN ON "RELOCATION"
 (1)  042140  010211                      MOV     R2,(R1)         ;LOAD 125253 USING ADDER  (PAR4 + VIRT ADDR.)
 (1)  042142  005067  135424              CLR     SR0             ;TURN OFF MEMORY MGMT.
 (1)  042146  011003                      MOV     (R0),R3         ;READ 125253 BACK WITHOUT USING MEM. MGMT.
 (1)  042150  016710  175416              MOV     $TMP0,(R0)      ;RESTORE ORIGINAL CONTENTS TO TEST LOC.
 (1)  042154  020203                      CMP     R2,R3           ;WAS SAME PATTERN READ BACK THAT WAS
 (1)                                                          ;WRITTEN USING "DEST-ONLY-RELOC."?
 (2)  042156  001401                      BEQ     3$
 (3)  042160  104000                      EMT                     ;TEST LOCATION DID NOT HAVE PATTERN
 (1)                                                          ;THAT SHOULD HAVE BEEN WRITTEN TO IT.
```

```
  (1)                                                                  ;APPARENTLY PHYSICAL ADDR. WAS
  (1)                                                                  ;FORMED WRONG BY ADDERS USING
  (1)                                                                  ;THE VIRTUAL ADDR. AND KIPAR4
  (1)                                                                  ;FOR TIGHTER SCOPE LOOP
  (1)                                                                  ;REPLACE ERROR CALL WITH
  (1)                                                                  ;"BR 2$" = 000742
  (1)  042162                          3$:
12910  042162  012700  062276          4$:     MOV     #62276,R0       ;LOAD PHYSICAL ADDR. PBA INTO R0
  (1)  042166  012701  107376                  MOV     #107376,R1      ;LOAD VIRTUAL ADDR. VBA INTO R1
  (1)  042172  012702  125254                  MOV     #125254,R2      ;LOAD TEST PATTERN INTO R2
  (1)  042176  012704  000527                  MOV     #527,R4 ;LOAD R4 WITH PAR VALUE
  (1)  042202  010467  130142                  MOV     R4,KIPAR4       ;LOAD KERNEL PAR 4 BITS <11:00>
  (1)  042206  011067  175360                  MOV     (R0),$TMP0      ;SAVE CONTENTS AT TEST LOCATION
  (1)  042212  052767  000020  130276          BIS     #BIT4,SR3       ;SET UP FOR 22-BIT ADDRESSING
  (1)  042220  052767  000001  135344          BIS     #BIT0,SR0       ;TURN ON "RELOCATION"
  (1)  042226  010211                          MOV     R2,(R1)         ;LOAD 125254 USING ADDER  (PAR4 + VIRT ADDR.)
  (1)  042230  005067  135336                  CLR     SR0             ;TURN OFF MEMORY MGMT.
  (1)  042234  011003                          MOV     (R0),R3         ;READ 125254 BACK WITHOUT USING MEM. MGMT.
  (1)  042236  016710  175330                  MOV     $TMP0,(R0)      ;RESTORE ORIGINAL CONTENTS TO TEST LOC.
  (1)  042242  020203                          CMP     R2,R3           ;WAS SAME PATTERN READ BACK THAT WAS
  (1)                                                                  ;WRITTEN USING "DEST-ONLY-RELOC."?
  (2)  042244  001401                          BEQ     5$
  (3)  042246  104000                          EMT                     ;TEST LOCATION DID NOT HAVE PATTERN
  (1)                                                                  ;THAT SHOULD HAVE BEEN WRITTEN TO IT.
  (1)                                                                  ;APPARENTLY PHYSICAL ADDR. WAS
  (1)                                                                  ;FORMED WRONG BY ADDERS USING
  (1)                                                                  ;THE VIRTUAL ADDR. AND KIPAR4
  (1)                                                                  ;FOR TIGHTER SCOPE LOOP
  (1)                                                                  ;REPLACE ERROR CALL WITH
  (1)                                                                  ;"BR 4$" = 000742
  (1)  042250                          5$:
12911  042250  012700  062076          6$:     MOV     #62076,R0       ;LOAD PHYSICAL ADDR. PBA INTO R0
  (1)  042254  012701  104576                  MOV     #104576,R1      ;LOAD VIRTUAL ADDR. VBA INTO R1
  (1)  042260  012702  125255                  MOV     #125255,R2      ;LOAD TEST PATTERN INTO R2
  (1)  042264  012704  000553                  MOV     #553,R4 ;LOAD R4 WITH PAR VALUE
  (1)  042270  010467  130054                  MOV     R4,KIPAR4       ;LOAD KERNEL PAR 4 BITS <11:00>
  (1)  042274  011067  175272                  MOV     (R0),$TMP0      ;SAVE CONTENTS AT TEST LOCATION
  (1)  042300  052767  000020  130210          BIS     #BIT4,SR3       ;SET UP FOR 22-BIT ADDRESSING
  (1)  042306  052767  000001  135256          BIS     #BIT0,SR0       ;TURN ON "RELOCATION"
  (1)  042314  010211                          MOV     R2,(R1)         ;LOAD 125255 USING ADDER  (PAR4 + VIRT ADDR.)
  (1)  042316  005067  135250                  CLR     SR0             ;TURN OFF MEMORY MGMT.
  (1)  042322  011003                          MOV     (R0),R3         ;READ 125255 BACK WITHOUT USING MEM. MGMT.
  (1)  042324  016710  175242                  MOV     $TMP0,(R0)      ;RESTORE ORIGINAL CONTENTS TO TEST LOC.
  (1)  042330  020203                          CMP     R2,R3           ;WAS SAME PATTERN READ BACK THAT WAS
  (1)                                                                  ;WRITTEN USING "DEST-ONLY-RELOC."?
  (2)  042332  001401                          BEQ     7$
  (3)  042334  104000                          EMT                     ;TEST LOCATION DID NOT HAVE PATTERN
  (1)                                                                  ;THAT SHOULD HAVE BEEN WRITTEN TO IT.
  (1)                                                                  ;APPARENTLY PHYSICAL ADDR. WAS
  (1)                                                                  ;FORMED WRONG BY ADDERS USING
  (1)                                                                  ;THE VIRTUAL ADDR. AND KIPAR4
  (1)                                                                  ;FOR TIGHTER SCOPE LOOP
  (1)                                                                  ;REPLACE ERROR CALL WITH
  (1)                                                                  ;"BR 6$" = 000742
  (1)  042336                          7$:
12912  042336  012700  000000          8$:     MOV     #000000,R0      ;LOAD PHYSICAL ADDR. PBA INTO R0
```

```
  (1)  042342  012701  111400                  MOV     #111400,R1       ;LOAD VIRTUAL ADDR. VBA INTO R1
  (1)  042346  012702  125256                  MOV     #125256,R2       ;LOAD TEST PATTERN INTO R2
  (1)  042352  012704  177664                  MOV     #177664,R4       ;LOAD R4 WITH PAR VALUE
  (1)  042356  010467  127766                  MOV     R4,KIPAR4        ;LOAD KERNEL PAR 4 BITS <11:00>
  (1)  042362  011067  175204                  MOV     (R0),$TMP0       ;SAVE CONTENTS AT TEST LOCATION
  (1)  042366  052767  000020  130122          BIS     #BIT4,SR3        ;SET UP FOR 22-BIT ADDRESSING
  (1)  042374  052767  000001  135170          BIS     #BIT0,SR0        ;TURN ON "RELOCATION"
  (1)  042402  010211                          MOV     R2,(R1)          ;LOAD 125256 USING ADDER  (PAR4 + VIRT ADDR.)
  (1)  042404  005067  135162                  CLR     SR0              ;TURN OFF MEMORY MGMT.
  (1)  042410  011003                          MOV     (R0),R3          ;READ 125256 BACK WITHOUT USING MEM. MGMT.
  (1)  042412  016710  175154                  MOV     $TMP0,(R0)       ;RESTORE ORIGINAL CONTENTS TO TEST LOC.
  (1)  042416  020203                          CMP     R2,R3            ;WAS SAME PATTERN READ BACK THAT WAS
  (1)                                                                   ;WRITTEN USING "DEST-ONLY-RELOC."?
  (2)  042420  001401                          BEQ     9$
  (3)  042422  104000                          EMT                      ;TEST LOCATION DID NOT HAVE PATTERN
  (1)                                                                   ;THAT SHOULD HAVE BEEN WRITTEN TO IT.
  (1)                                                                   ;APPARENTLY PHYSICAL ADDR. WAS
  (1)                                                                   ;FORMED WRONG BY ADDERS USING
  (1)                                                                   ;THE VIRTUAL ADDR. AND KIPAR4
  (1)                                                                   ;FOR TIGHTER SCOPE LOOP
  (1)                                                                   ;REPLACE ERROR CALL WITH
  (1)                                                                   ;"BR 8$" = 000742
  (1)  042424                          9$:
12913
12937                                  ;*******************************************************************************
  (2)                                  ;TEST 375       READ AND WRITE WHILE IN RELOCATE MODE
  (3)                                  ;*******************************************************************************
  (2)  042424                          TS375:
12938
12939  042424  005067  135346          1$:     CLR     PSW              ;START IN KERNEL MODE
12940  042430  012704  001377                  MOV     #1377,R4         ;LOAD R4 WITH VALUE FOR PAR4
12941  042434  012705  001400                  MOV     #1400,R5         ;LOAD R5 WITH VALUE FOR PAR5
12942  042440  010467  127704                  MOV     R4,KIPAR4        ;LOAD KERNEL PAR4
12943  042444  010567  127702                  MOV     R5,KIPAR5        ;LOAD KERNEL PAR5
12944  042450  012700  177640                  MOV     #UIPAR0,R0       ;LOAD ADDRESS OF FIRST USER PAR IN R0
12945  042454  005001                          CLR     R1               ;CLEAR R1
12946  042456  012702  000007                  MOV     #7,R2            ;LOAD LOOP COUNTER WITH A 7
12947  042462  010120                  2$:     MOV     R1,(R0)+         ;MAP USER PAR'S TO PAGES 0-6 (4K EACH)
12948  042464  062701  000200                  ADD     #200,R1
12949  042470  077204                          SOB     R2,2$            ;LOOP UNTIL UIPAR0-UIPAR6 ARE LOADED
12950  042472  012710  177600                  MOV     #177600,(R0)     ;MAP USER PAR7 TO THE I/O PAGE
12951  042476  012700  177600                  MOV     #UIPDR0,R0       ;LOAD ADDRESS OF FIRST USER PDR IN R0
12952  042502  012701  077406                  MOV     #77406,R1        ;LOAD PDR DATA INTO R1
12953  042506  012702  000010                  MOV     #10,R2           ;LOAD LOOP COUNTER WITH AN 8
12954  042512  010120                  3$:     MOV     R1,(R0)+         ;MAP ALL 8 PAGES 128 BLOCKS, UPWARD
12955  042514  077202                          SOB     R2,3$            ;       EXPANDABLE, READ/WRITE
12956  042516  012767  042762  135524          MOV     #8$,MMVEC        ;SET M. M. TRAP VECTOR TO 8$
12957  042524  052767  000020  127764          BIS     #BIT4,SR3        ;SET UP FOR 22-BIT ADDRESSING
12958  042532  012767  000001  135032          MOV     #BIT0,SR0        ;TURN ON MEMORY MANAGEMENT
12959  042540  105067  135044                  CLRB    UIPDR4           ;MAP USER SPACE NON-RESIDENT WHILE
12960  042544  105067  135042                  CLRB    UIPDR5           ;       TESTING KERNEL SPACE
12961  042550  010567  135074                  MOV     R5,UIPAR4        ;MAP USER PAR'S OPPOSITE OF KIPAR'S
12962  042554  010467  135072                  MOV     R4,UIPAR5
12963  042560  016767  135212  175004  4$:     MOV     PSW,$TMP0        ;SAVE PSW IN CASE OF ERROR
12964  042566  012700  100100                  MOV     #100100,R0       ;PUT VIRTUAL ADDR. THAT USES PAR4 IN R0
12965  042572  012701  120000                  MOV     #120000,R1       ;PUT VIRTUAL ADDR. THAT USES PAR5 IN R1
```

```
042576  010010                  5$:     MOV     R0,(R0)         ;WRITE TO TEST LOC. USING PAR4
042600  011102                          MOV     (R1),R2         ;READ THE SAME LOC., BUT USING PAR5
042602  020002                          CMP     R0,R2           ;DID WE READ WHAT WE WROTE?
042604  001401                          BEQ     6$
042606  104000                          EMT                     ;READING LOC. USING PAR5 AND A VIRT.
                                                                ;ADDR. DID NOT FIND DATA WRITTEN WHEN USING
                                                                ;PAR4 AND VIRT. ADDRESS.
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;"BR 5$" = 000765
042610  062700  000100          6$:     ADD     #100,R0         ;CHANGE VIRTUAL ADDRS. TO POINT TO NEXT BLOCK
042614  062701  000100                  ADD     #100,R1
042620  020127  127700                  CMP     R1,#127700      ;WERE BLOCKS FROM 60000-676000 ALL TRIED?
042624  001364                          BNE     5$              ;BRANCH IF NO
042626  032767  140000  135142          BIT     #140000,PSW     ;HAVE WE DONE TEST IN USER MODE YET?
042634  001026                          BNE     7$              ;BRANCH IF YES
042636  010467  135006                  MOV     R4,UIPAR4       ;LOAD USER PAR4
042642  010567  135004                  MOV     R5,UIPAR5       ;LOAD USER PAR5
042646  112767  000006  134734          MOVB    #6,UIPDR4       ;MAP USER SPACE R/W TO TEST IT
042654  112767  000006  134730          MOVB    #6,UIPDR5
042662  105067  127422                  CLRB    KIPDR4          ;MAP  KERNEL SPACE NON-RESIDENT WHILE
042666  105067  127420                  CLRB    KIPDR5          ;       TESTING USER SPACE
042672  010567  127452                  MOV     R5,KIPAR4       ;MAP KERNEL PAR'S OPPOSITE UIPAR'S
042676  010467  127450                  MOV     R4,KIPAR5
042702  012767  140000  135066          MOV     #140000,PSW     ;GO TO USER MODE
042710  000723                          BR      4$              ;GO BACK AND READ/WRITE IN USER MODE
042712  005067  135060          7$:     CLR     PSW             ;GO BACK TO KERNEL MODE BEFORE LEAVING
042716  012767  077406  127364          MOV     #77406,KIPDR4   ;REMAP KERNEL PAGES READ/WRITE
042724  012767  077406  127360          MOV     #77406,KIPDR5
042732  010567  127412                  MOV     R5,KIPAR4       ;MAP KERNEL  AND USER PAR'S 4 & 5
042736  010567  127410                  MOV     R5,KIPAR5       ;  BACK TO 12-16K
042742  010567  134702                  MOV     R5,UIPAR4
042746  010567  134700                  MOV     R5,UIPAR5
042752  012767  021550  135270          MOV     #T0250,MMVEC    ;RESTORE ADDR. OF NORMAL M.M. TRAP ROUTINE
042760  000404                          BR      TS376           ;GET TO NEXT TEST
042762  042767  160000  134602  8$:     BIC     #160000,SR0     ;CLEAR ERROR BITS IN SR0
042770  104000                          EMT                     ;M.M. TRAP WHILE IN RELOCATE MODE -
                                                                ;REFERENCED WRONG SET OF PDR'S
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;A "NOP" = 000240

                                ;*******************************************************************************
                                ;TEST 376       W-BIT LOGIC TEST, KERNEL PDR'S
                                ;*******************************************************************************
042772                          TS376:
042772                          1$:
042772  004767  005262                  JSR     PC,TOFF         ;TURN T-BIT TRAPPING OFF FOR THIS TEST
042776  012702  000004                  MOV     #4,R2           ;SET LOOP COUNTER TO 4
043002  012700  172346                  MOV     #KIPAR3,R0      ;LOAD ADDRESS OF PAR3 INTO R0
043006  012701  001400                  MOV     #1400,R1        ;LOAD "24-28K" PAR VALUE INTO R1
043012  010120                  2$:     MOV     R1,(R0)+        ;MAP PARS 3-6 TO 12-16K
043014  077202                          SOB     R2,2$           ;LOOP TIL ALL 4 OF THEM LOADED
043016  012705  172300                  MOV     #KIPDR0,R5      ;LOAD ADDRESS OF FIRST PDR TO BE TESTED IN R5
043022  012704  000010                  MOV     #10,R4          ;SET LOOP COUNTER TO 8
043026  012703  017776                  MOV     #17776,R3       ;INITIALIZE VIRTUAL ADDRESS TO BE IN R3
```

```
  (1)  043032  012700  172300          3$:   MOV   #KIPDR0,R0      ;LOAD ADDR. OF FIRST PDR TO BE SETUP IN R0
  (1)  043036  012702  000010                MOV   #10,R2           ;SET LOOP COUNTER TO 8
  (1)  043042  012701  077406                MOV   #77406,R1        ;PUT "W-BIT OFF DATA" INTO R1
  (1)  043046  010120                 4$:    MOV   R1,(R0)+         ;CLEAR ALL W-BITS BY WRITING TO ALL PDRS
  (1)  043050  077202                        SOB   R2,4$            ;LOOP UNTIL ALL OF THEM SETUP
  (1)  043052  011313                        MOV   (R3),(R3)        ;DO "DATO" TO VIRTUAL ADDR.-SETTING A W-BIT
  (1)  043054  031527  000100                BIT   (R5),#WBIT       ;DID THAT CAUSE W-BIT TO BE SET?
  (2)  043060  001001                        BNE   5$
  (3)  043062  104000                        EMT                    ;W-BIT DID NOT GET SET IN PDR
  (1)                                                               ;FOR TIGHTER SCOPE LOOP
  (1)                                                               ;REPLACE ERROR CALL WITH
  (1)                                                               ;"BR 3$" = 000763
  (1)  043064  012702  000010          5$:   MOV   #10,R2           ;SET LOOP COUNTER TO 8
  (1)  043070  012700  172300                MOV   #KIPDR0,R0       ;LOAD ADDR. OF FIRST PDR TO BE CHECKED IN R0
  (1)  043074  031027  000100          6$:   BIT   (R0),#WBIT       ;DID W-BIT IN OTHER PDRS REMAIN CLEAR?
  (1)  043100  001403                        BEQ   7$               ;BRANCH IF YES
  (1)  043102  020500                        CMP   R5,R0            ;IF W-BIT SET, THEN WAS IT PDR UNDER TEST?
  (2)  043104  001401                        BEQ   7$
  (3)  043106  104000                        EMT                    ;W-BIT GOT SET IN MORE THAN ONE PDR
  (1)                                                               ;FOR TIGHTER SCOPE LOOP
  (1)                                                               ;REPLACE ERROR CALL WITH
  (1)                                                               ;"BR 3$" = 000750
  (1)  043110  062700  000002          7$:   ADD   #2,R0            ;POINT R0 TO NEXT PDR TO BE CHECKED
  (1)  043114  077211                        SOB   R2,6$            ;LOOP UNTIL ALL 8 CHECKED FOR CLEAR W-BIT
  (1)  043116  010115                        MOV   R1,(R5)          ;WRITE TO THE PDR TESTED TO CLEAR W-BIT
  (1)  043120  031527  000100                BIT   (R5),#WBIT       ;DID WRITING PDR CLEAR THE W-BIT?
  (2)  043124  001401                        BEQ   8$
  (3)  043126  104000                        EMT                    ;W-BIT DID NOT CLEAR BY WRITING THE PDR
  (1)                                                               ;FOR TIGHTER SCOPE LOOP
  (1)                                                               ;REPLACE ERROR CALL WITH
  (1)                                                               ;"BR 3$" = 000740
  (1)  043130  062705  000002          8$:   ADD   #2,R5            ;POINT R5 TO THE NEXT PDR TO BE TESTED
  (1)  043134  062703  020000                ADD   #20000,R3        ;CHANGE VIRT. ADDR TO REF. NEXT PDR
  (1)  043140  077444                        SOB   R4,3$            ;LOOP BACK TO 3$ UNTIL ALL 8 PDR'S TESTED
  (1)  043142  004767  005146                JSR   PC,TON           ;TURN T-BIT BACK ON FOR NEXT TEST
13012
13016                                       ;*************************************************************************************
  (2)                                       ;TEST 377       W-BIT LOGIC TEST, USER PDR'S
  (3)                                       ;*************************************************************************************
  (2)  043146                               TS377:
13017  043146  012767  140000  134622  1$:   MOV   #140000,PSW      ;GO TO USER MODE FOR THIS TEST
13018  043154  004767  005100                JSR   PC,TOFF          ;TURN T-BIT TRAPPING OFF FOR THIS TEST
  (1)  043160  012702  000004                MOV   #4,R2            ;SET LOOP COUNTER TO 4
  (1)  043164  012700  177646                MOV   #UIPAR3,R0       ;LOAD ADDRESS OF PAR3 INTO R0
  (1)  043170  012701  001400                MOV   #1400,R1         ;LOAD "24-28K" PAR VALUE INTO R1
  (1)  043174  010120                 2$:    MOV   R1,(R0)+         ;MAP PARS 3-6 TO 12-16K
  (1)  043176  077202                        SOB   R2,2$            ;LOOP TIL ALL 4 OF THEM LOADED
  (1)  043200  012705  177600                MOV   #UIPDR0,R5       ;LOAD ADDRESS OF FIRST PDR TO BE TESTED IN R5
  (1)  043204  012704  000010                MOV   #10,R4           ;SET LOOP COUNTER TO 8
  (1)  043210  012703  017776                MOV   #17776,R3        ;INITIALIZE VIRTUAL ADDRESS TO BE IN R3
  (1)  043214  012700  177600          3$:   MOV   #UIPDR0,R0       ;LOAD ADDR. OF FIRST PDR TO BE SETUP IN R0
  (1)  043220  012702  000010                MOV   #10,R2           ;SET LOOP COUNTER TO 8
  (1)  043224  012701  077406                MOV   #77406,R1        ;PUT "W-BIT OFF DATA" INTO R1
  (1)  043230  010120                 4$:    MOV   R1,(R0)+         ;CLEAR ALL W-BITS BY WRITING TO ALL PDRS
  (1)  043232  077202                        SOB   R2,4$            ;LOOP UNTIL ALL OF THEM SETUP
  (1)  043234  011313                        MOV   (R3),(R3)        ;DO "DATO" TO VIRTUAL ADDR.-SETTING A W-BIT
```

```
  (1) 043236 031527 000100          BIT   (R5),#WBIT     ;DID THAT CAUSE W-BIT TO BE SET?
  (2) 043242 001001                 BNE   5$
  (3) 043244 104000                 EMT                   ;W-BIT DID NOT GET SET IN PDR
  (1)                                                     ;FOR TIGHTER SCOPE LOOP
  (1)                                                     ;REPLACE ERROR CALL WITH
  (1)                                                     ;"BR 3$" = 000763
  (1) 043246 012702 000010          5$:   MOV   #10,R2    ;SET LOOP COUNTER TO 8
  (1) 043252 012700 177600          MOV   #UIPDR0,R0      ;LOAD ADDR. OF FIRST PDR TO BE CHECKED IN R0
  (1) 043256 031027 000100          6$:   BIT   (R0),#WBIT ;DID W-BIT IN OTHER PDRS REMAIN CLEAR?
  (1) 043262 001403                 BEQ   7$              ;BRANCH IF YES
  (1) 043264 020500                 CMP   R5,R0           ;IF W-BIT SET, THEN WAS IT PDR UNDER TEST?
  (2) 043266 001401                 BEQ   7$
  (3) 043270 104000                 EMT                   ;W-BIT GOT SET IN MORE THAN ONE PDR
  (1)                                                     ;FOR TIGHTER SCOPE LOOP
  (1)                                                     ;REPLACE ERROR CALL WITH
  (1)                                                     ;"BR 3$" = 000750
  (1) 043272 062700 000002          7$:   ADD   #2,R0     ;POINT R0 TO NEXT PDR TO BE CHECKED
  (1) 043276 077211                 SOB   R2,6$           ;LOOP UNTIL ALL 8 CHECKED FOR CLEAR W-BIT
  (1) 043300 010115                 MOV   R1,(R5)         ;WRITE TO THE PDR TESTED TO CLEAR W-BIT
  (1) 043302 031527 000100          BIT   (R5),#WBIT      ;DID WRITING PDR CLEAR THE W-BIT?
  (2) 043306 001401                 BEQ   8$
  (3) 043310 104000                 EMT                   ;W-BIT DID NOT CLEAR BY WRITING THE PDR
  (1)                                                     ;FOR TIGHTER SCOPE LOOP
  (1)                                                     ;REPLACE ERROR CALL WITH
  (1)                                                     ;"BR 3$" = 000740
  (1) 043312 062705 000002          8$:   ADD   #2,R5     ;POINT R5 TO THE NEXT PDR TO BE TESTED
  (1) 043316 062703 020000          ADD   #20000,R3       ;CHANGE VIRT. ADDR TO REF. NEXT PDR
  (1) 043322 077444                 SOB   R4,3$           ;LOOP BACK TO 3$ UNTIL ALL 8 PDR'S TESTED
  (1) 043324 004767 004764          JSR   PC,TON          ;TURN T-BIT BACK ON FOR NEXT TEST
13019 043330 005067 134442          CLR   PSW             ;BACK TO KERNEL MODE BEFORE LEAVING
13020
13030                               ;*************************************************************************
  (2)                               ;TEST 400       TEST "W-BIT" SPECIAL CASES
  (3)                               ;*************************************************************************
  (2) 043334                        TS400:
13031
13032 043334 004767 004720          1$:   JSR   PC,TOFF   ;TURN OFF T-BIT TRAPPING FOR THIS TEST
13033 043340 012701 077406          MOV   #77406,R1       ;PUT "W-BIT OFF" VALUE FOR PDR IN R1
13034 043344 010167 126746          2$:   MOV   R1,KIPDR7 ;LOAD KERNEL PDR 7 TO CLEAR W-BIT
13035 043350 016700 134216          MOV   SR0,R0          ;READ PRESENT CONTENTS OF STATUS REG. 0
13036 043354 010067 134212          MOV   R0,SR0          ;WRITE PRESENT CONTENTS OF SR0 BACK TO ITSELF
13037 043360 016702 126732          MOV   KIPDR7,R2       ;READ CONTENTS OF KIPDR7 INTO R2
13038 043364 020102                 CMP   R1,R2           ;WAS W-BIT LEFT CLEARED?
13039 043366 001401                 BEQ   3$
  (2) 043370 104000                 EMT                   ;W-BIT IN KIPDR7 SET WHEN SR0 WAS WRITTEN TO
13040                                                     ;FOR TIGHTER SCOPE LOOP
13041                                                     ;REPLACE ERROR CALL WITH
13042                                                     ;"BR 2$" = 000765
13043 043372 010167 126716          3$:   MOV   R1,KIPDR6 ;LOAD KERNEL PDR6 WITH 77406 TO CLEAR W-BIT
13044 043376 012767 043410 134400   MOV   #4$,ERRVEC      ;SET UP LOC. 4 TO 4$ FOR ODD ADDR. ABORT
13045 043404 005037 140000          CLR   @#140000        ;CAUSE TIMEOUT ABORT THRU LOC. 4
13046 043410 012706 001000          4$:   MOV   #KERSTK,SP ;RESTORE THE STACK POINTER
13047 043414 016702 126674          MOV   KIPDR6,R2       ;READ KIPDR6 INTO R2
13048 043420 052701 000100          BIS   #100,R1         ;R1-77506
13049 043424 020102                 CMP   R1,R2           ;WAS W-BIT SET?
13050 043426 001401                 BEQ   5$
```

```
  (2)  043430  104000                          EMT                     ;W-BIT WAS NOT SET DURING A TIMEOUT ABORT
13051                                                                   ;FOR TIGHTER SCOPE LOOP
13052                                                                   ;REPLACE ERROR CALL WITH
13053                                                                   ;"BR 3$" = 000757
13054  043432  010167  126656                5$:     MOV     R1,KIPDR6       ;RESTORE KIPDR6 TO 77406
13055  043436  012767  001400  126710               MOV     #1400,KIPAR6    ;RESTORE KIPAR6 TO 1400
13056  043444  012767  021526  134332               MOV     #T04,ERRVEC     ;RESTORE NORMAL CPU TRAP ROUTINE TO LOC.4
13057  043452  004767  004636                       JSR     PC,TON          ;TURN T-BIT TRAPPING BACK ON
13058
13059                                  ;*************************************************************************
13060                                  ;*
13061                                  ;*      THE NEXT THREE (3) TESTS CAUSE MEMORY MANAGEMENT ERRORS
13062                                  ;*      TO CHECK THE ABILITY OF STATUS REGISTER 0 TO RECORD KT
13063                                  ;*      ERRORS AND THE ABILITY OF STATUS REGISTER 2 TO LOCK UP THE
13064                                  ;*      VIRTUAL ADDR. OF THE INSTRUCTION THAT CAUSED THE ERROR.
13065                                  ;*      THE BITS OF SR2 ARE CHECKED AND BITS <15:13>, <6:5>, AND <3:0>
13066                                  ;*      ARE CHECKED IN SR0.  SO THE SR0 AND SR2 LOGIC AND THE
13067                                  ;*      KT ERROR LOGIC ARE CHECKED.
13068                                  ;*
13069                                  ;*************************************************************************
13070
13079                                  ;*************************************************************************
  (2)                                 ;TEST 401       NON-RESIDENT ABORT TEST (ACF=0E4)
  (3)                                 ;*************************************************************************
  (2)  043456                         TS401:
13080
13081  043456  012700  001400                1$:     MOV     #1400,R0        ;LOAD DATA FOR PAR'S INTO R0
13082  043462  010067  126660                        MOV     R0,KIPAR3       ;MAP KERNEL PAR'S 3&4 TO 24-28K
13083  043466  010067  126656                        MOV     R0,KIPAR4
13084  043472  010067  134150                        MOV     R0,UIPAR3       ;MAP USER PAR'S 3&4 TO 24-28K
13085  043476  010067  134146                        MOV     R0,UIPAR4
13086  043502  012767  077406  126576                MOV     #77406,KIPDR3   ;MAP KERNEL PDR 3  128 BLKS, READ-WRITE
13087  043510  012767  077406  134070                MOV     #77406,UIPDR3   ;MAP USER PDR 3  128 BLKS, READ-WRITE
13088  043516  012700  060000                        MOV     #60000,R0       ;LOAD VIRTUAL ADDR. TO REFERENCE PDR3 INTO R0
13089  043522  012701  100000                        MOV     #100000,R1      ;LOAD VIRTUAL ADDR. TO REFERENCE PDR4 INTO R1
13090  043526  012703  100011                        MOV     #100011,R3      ;LOAD R3 WITH WHAT SR0 SHOULD READ - N.R., KERNEL, PG.4
13091  043532  012702  077400                        MOV     #77400,R2       ;LOAD ACF=0 (NON-RESIDENT) PDR VALUE IN R2
13092  043536  012767  043572  134504        2$:     MOV     #5$,MMVEC       ;POINT MEM. MGMT. TRAP VECTOR TO 5$ BELOW
13093  043544  010267  126540                        MOV     R2,KIPDR4       ;LOAD ACF TEST VALUE INTO KIPDR4
13094  043550  010267  134034                        MOV     R2,UIPDR4       ;LOAD ACF TEST VALUE INTO UIPDR4
13095  043554  005010                        3$:     CLR     (R0)            ;CLEAR PHYS. LOC. 140000 USING PDR3
13096  043556  016767  134214  174006                MOV     PSW,$TMP0       ;SAVE PSW IN CASE OF ERROR
13097  043564  005211                        4$:     INC     (R1)            ;TRY TO REF. IT USING PDR4 - SHOULD TRAP TO 5$
13098  043566  001462                                BEQ     TS402
  (3)  043570  104000                                EMT                     ;MEM. MGMT. ABORT DID NOT OCCUR
13099                                                                   ;FOR TIGHTER SCOPE LOOP
13100                                                                   ;REPLACE ERROR CALL WITH
13101                                                                   ;"BR 3$" = 000772
13102  043572  062706  000004        5$:     ADD     #4,SP           ;RESTORE STACK POINTER
13103  043576  005710                        TST     (R0)            ;DID INSTRUCTION GET ABORTED & NOT EXECUTE
13104  043600  001401                        BEQ     6$
  (2)  043602  104000                        EMT                     ;INSTRUCTION WAS NOT ABORTED, LOC. GOT CHANGED
13105                                                                   ;FOR TIGHTER SCOPE LOOP
13106                                                                   ;REPLACE ERROR CALL WITH
13107                                                                   ;"BR 3$" = 000764
13108  043604  016767  133762  173752  6$:     MOV     SR0,WASSR0      ;READ STATUS REGISTER 0
```

```
13109  043612  016767  133760  173746          MOV     SR2,WASSR2      ;READ STATUS REGISTER 2
13110  043620  020367  173740                  CMP     R3,WASSR0       ;DID SR0 REPORT NON-RESIDENT ERROR CORRECTLY?
13111  043624  001401                          BEQ     7$
  (2)  043626  104000                          EMT                     ;SR0 DID NOT REPORT NON-RES. ERROR CORRECTLY
13112                                                                  ;FOR TIGHTER SCOPE LOOP
13113                                                                  ;REPLACE ERROR CALL WITH
13114                                                                  ;"BR 3$" = 000752
13115  043630  012704  043564          7$:     MOV     #4$,R4          ;LOAD R4 WITH WHAT SR2 SHOULD READ
13116  043634  020467  173726                  CMP     R4,WASSR2       ;DID SR2 LOCKUP RIGHT VIRTUAL ADDR. (=4$)?
13117  043640  001401                          BEQ     8$
  (2)  043642  104000                          EMT                     ;SR2 DID NOT LOCK VIRTUAL ADDR. OF NON-RES. ERROR
13118                                                                  ;FOR TIGHTER SCOPE LOOP
13119                                                                  ;REPLACE ERROR CALL WITH
13120                                                                  ;"BR 3$" = 000744
13121  043644  042767  160000  133720  8$:     BIC     #160000,SR0     ;CLEAR THE ERROR BITS IN SR0
13122  043652  032767  140000  173712          BIT     #140000,$TMP0   ;HAS ACF=064 BEEN TESTED IN USER YET
13123  043660  001006                          BNE     9$              ;BRANCH IF YES
13124  043662  012703  100151                  MOV     #100151,R3      ;LOAD R3 WITH WHAT SR0 SHOULD READ - N.R., USER, PG.4
13125  043666  012767  140000  134102          MOV     #140000,PSW     ;GO TO USER MODE
13126  043674  000720                          BR      2$              ;REPEAT TEST IN USER MODE
13127  043676  022702  077404          9$:     CMP     #77404,R2       ;HAS ACF=4 BEEN TESTED YET?
13128  043702  001407                          BEQ     10$             ;BRANCH IF YES
13129  043704  012702  077404                  MOV     #77404,R2       ;THEN LOAD ACF=4 (NON-RES) PDR VALUE IN R2
13130  043710  012703  100011                  MOV     #100011,R3      ;LOAD R3 WITH WHAT SR0 SHOULD READ-N.R.,KERNEL,PG. 4
13131  043714  005067  134056                  CLR     PSW             ;GO BACK TO KERNEL MODE
13132  043720  000706                          BR      2$              ;GO BACK & TEST ACF=4 IN SAME MODE
13133  043722  005067  134050          10$:    CLR     PSW             ;GO BACK TO KERNEL MODE BEFORE LEAVING
13134  043726  012767  021550  134314          MOV     #T0250,MMVEC    ;RESTORE ADDRESS OF NORMAL MEMORY
13135                                                                  ;MANAGEMENT ERROR ROUTINE TO MMVEC
13136
13145                          ;*****************************************************************************
  (2)                          ;TEST 402       READ-ONLY ABORT TEST (ACF=2)
  (3)                          ;*****************************************************************************
  (2)  043734                  TS402:
13146  043734                  1$:                                     ;KERNEL & USER PAR'S 3 & 4 AND PDR 3
13147                                                                  ;ARE SETUP FROM LAST TEST
13148  043734  012700  060000          MOV     #60000,R0               ;LOAD VIRTUAL ADDR. TO REFERENCE PDR3 INTO R0
13149  043740  012701  100000          MOV     #100000,R1              ;LOAD VIRTUAL ADDR. TO REFERENCE PDR4 INTO R1
13150  043744  012703  020011          MOV     #20011,R3               ;LOAD R3 WITH WHAT SR0 SHOULD READ - R/O, KERNEL, PG.4
13151  043750  012702  077402          MOV     #77402,R2               ;LOAD ACF=2 (READ-ONLY) PDR VALUE IN R2
13152  043754  012767  044006  134266  2$:     MOV     #5$,MMVEC       ;POINT MEM. MGMT. TRAP VECTOR TO 5$ BELOW
13153  043762  010267  126322          MOV     R2,KIPDR4               ;LOAD ACF=2 INTO KIPDR4
13154  043766  010267  133616          MOV     R2,UIPDR4               ;LOAD ACF=2 INTO UIPDR4
13155  043772  005010                  3$:     CLR     (R0)            ;CLEAR PHYS. LOC. 140000 USING PDR3
13156  043774  016767  133776  173570          MOV     PSW,$TMP0       ;SAVE PSW IN CASE OF ERROR
13157  044002  005211                  4$:     INC     (R1)            ;TRY TO WRITE USING PDR4 - SHOULD TRAP TO 5$
13158  044004  104000                          EMT                     ;MEM. MGMT. ABORT DID NOT OCCUR
13159                                                                  ;FOR TIGHTER SCOPE LOOP
13160                                                                  ;REPLACE ERROR CALL WITH
13161                                                                  ;"BR 3$" = 000772
13162  044006  062706  000004          5$:     ADD     #4,SP           ;RESTORE STACK POINTER
13163  044012  005710                          TST     (R0)            ;DID INSTRUCTION GET ABORTED & NOT EXECUTE
13164  044014  001401                          BEQ     6$
  (2)  044016  104000                          EMT                     ;INSTRUCTION WAS NOT ABORTED, LOC. GOT CHANGED
13165                                                                  ;FOR TIGHTER SCOPE LOOP
13166                                                                  ;REPLACE ERROR CALL WITH
```

```
13167                                                                   ;"BR 3$" = 000764
13168  044020  016767  133546  173536  6$:     MOV     SR0,WASSR0      ;READ STATUS REG. 0
13169  044026  016767  133544  173532          MOV     SR2,WASSR2      ;READ STATUS REG. 2
13170  044034  020367  173524                  CMP     R3,WASSR0       ;DID SR0 REPORT READ-ONLY ERROR CORRECTLY?
13171  044040  001401                          BEQ     7$
  (2)  044042  104000                          EMT                     ;SR0 DID NOT REPORT R/O ERROR CORRECTLY
13172                                                                   ;FOR TIGHTER SCOPE LOOP
13173                                                                   ;REPLACE ERROR CALL WITH
13174                                                                   ;"BR 3$" = 000752
13175  044044  012704  044002          7$:     MOV     #4$,R4          ;LOAD R4 WITH WHAT SR2 SHOULD READ
13176  044050  020467  173512                  CMP     R4,WASSR2       ;DID SR2 LOCKUP RIGHT VIRTUAL ADDR. (=4$)?
13177  044054  001401                          BEQ     8$
  (2)  044056  104000                          EMT                     ;SR2 DID NOT LOCKUP VIRTUAL ADDR. OF R/O ERROR
13178                                                                   ;FOR TIGHTER SCOPE LOOP
13179                                                                   ;REPLACE ERROR CALL WITH
13180                                                                   ;'BR 3$" = 000744
13181  044060  042767  160000  133504  8$:     BIC     #160000,SR0     ;CLEAR THE ERROR BITS IN SR0
13182  044066  032767  140000  173476          BIT     #140000,$TMP0   ;HAS ACF=2 BEEN TESTED IN USER MODE?
13183  044074  001006                          BNE     9$              ;BRANCH IF YES
13184  044076  012703  020151                  MOV     #20151,R3       ;LOAD R3 WITH WHAT SR0 SHOULD READ-R/O, USER, PG.4
13185  044102  012767  140000  133666          MOV     #140000,PSW     ;GO TO USER MODE
13186  044110  000721                          BR      2$              ;REPEAT TEST IN USER MODE
13187  044112  005067  133660          9$:     CLR     PSW             ;GO BACK TO KERNEL MODE BEFORE LEAVING
13188  044116  012767  021550  134124          MOV     #T0250,MMVEC    ;RESTORE ADDRESS OF NORMAL MEMORY
13189                                                                   ;MANAGEMENT ERROR ROUTINE TO MMVEC.
13190
13191
13192
13193                           ;NOTE:  MACRO MSG31A WAS DELETED AS IT DIDN'T APPLY TO F11.
13194
13195
13196
13197
13198
13199
13200
13201
13202  ;*****************************************************************************************
13203  ;*
13204  ;*      THE NEXT TWO (2) TESTS WILL BE CHECKING THE PAGE LENGTH
13205  ;*      COMPARATORS AND SOME MORE OF THE KT ERROR DETECTION
13206  ;*      AND STATUS LOGIC.  THE PAGE LENGTH FIELD (PLF) IN KERNEL
13207  ;*      PDR 4 IS VARIED AND FOR EVERY PLF, THREE (3) VIRTUAL
13208  ;*      ADDRESSES ARE READ.  WHILE USING BOTH UPWARD & DOWNWARD PAGE
13209  ;*      EXPANSION, ONE OF THOSE THREE VIRTUAL ADDRESSES WILL CAUSE A
13210  ;*      "PAGE LENGTH ABORT" WHILE THE OTHER TWO WON'T.
13211  ;*
13212  ;*      STATUS REGISTER 0 & 2 ARE CHECKED WHEN THE PAGE LENGTH
13213  ;*      ABORT DOES OCCUR TO SEE THAT THE ABORT IS REPORTED AND THAT
13214  ;*      THE VIRTUAL ADDRESS OF THE INSTRUCTION THAT CAUSED THE ABORT
13215  ;*      IS LOCKED UP.
13216  ;*
13217  ;*****************************************************************************************
13218
13219
13231
```

```
                                        ;*******************************************************************************
                                        ;TEST 403        PAGE LENGTH FAULTS-UPWARD EXPANSION
                                        ;*******************************************************************************
044124                                  TS403:
044124  012767  077406  126154  1$:     MOV     #77406,KIPDR3   ;MAKE SURE PDR3 IS DESCRIBED AS R/W
044132  012767  077406  126152          MOV     #77406,KIPDR5   ;MAKE SURE PDR5 IS DESCRIBED AS R/W
044140  012700  044340                  MOV     #DALTB1,R0      ;DAL TABLE FOR VIRTUAL ADDR'S. TO SELECT PDR4.
044144  012704  044356                  MOV     #PDRTB1,R4      ;PDR TABLE FOR PDR4 (COINCIDES WITH DAL TABLE).
044150  012701  000006                  MOV     #6,R1           ;SET UP LOOP COUNTER.
044154  012767  044316  134066          MOV     #9$,MMVEC       ;SETUP M.M. TRAP VECTOR FOR UNEXPECTED ABORTS
044162  012706  001000                  MOV     #KERSTK,KSP     ;MAKE SURE STACK POINTER IS ALL SET UP

                                        ;TEST NON-ABORT CASES (VBA < OR = PLF)
044166  012467  126116          2$:     MOV     (R4)+,KIPDR4    ;LOAD KIPDR4 WITH PAGE LENGTH VALUE
044172  005730                          TST     @(R0)+          ;ACCESS VIRTUAL ADDR. (VBA < OR = PLF)
                                                                ;NO ABORT SHOULD OCCUR!!!
044174  077104                          SOB     R1,2$           ;DONE?...NO- TEST NEXT COMBINATION OF DAL & PDR.

                                        ;TEST ABORT CASES (VBA > PLF)
044176  012701  000005          3$:     MOV     #5,R1           ;SET UP LOOP COUNTER.
044202  012700  044374                  MOV     #DALTB2,R0      ;DAL TABLE
044206  012704  044410                  MOV     #PDRTB2,R4      ;PDR TABLE
044212  012767  044232  134030          MOV     #6$,MMVEC       ;SETUP M.M. TRAP VECTOR FOR EXPECTED ABORT

044220  012467  126064          4$:     MOV     (R4)+,KIPDR4    ;LOAD KIPDR4 WITH PAGE LENGHT VALUE
044224  005730                  5$:     TST     @(R0)+          ;ACCESS VIRTUAL ADDR. (VBA > PLF - ABORT TO 6$)
044226  001476                          BEQ     TS404
044230  104000                          EMT                     ;EXPECTED PAGE LENGTH ABORT DID NOT OCCUR
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;"BR 5$" = 000776
044232  012706  001000          6$:     MOV     #KERSTK,KSP     ;RESTORE STACK POINTER FOLLOWING ABORT
044236  016767  133330  173320          MOV     SR0,WASSR0      ;READ M.M. STATUS REG. 0
044244  016767  133326  173314          MOV     SR2,WASSR2      ;READ M.M. STATUS REG. 2
044252  012702  040011                  MOV     #40011,R2       ;PUT EXPECTED SR0 CONTENTS IN R2
044256  020267  173302                  CMP     R2,WASSR0       ;DID SR0 REPORT PG. LENGTH ABORT, PAGE 4, KERNEL?
044262  001401                          BEQ     7$
044264  104000                          EMT                     ;SR0 DID NOT REPORT PG. LENGTH ABORT CORRECTLY
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;"BR 5$" = 000757
044266  012703  044224          7$:     MOV     #5$,R3          ;PUT EXPECTED SR2 CONTENTS IN R3
044272  020367  173270                  CMP     R3,WASSR2       ;DID SR2 LOCKUP VIRT. ADDR. OF ABORTED INSTRUCTION?
044276  001401                          BEQ     8$
044300  104000                          EMT                     ;SR2 DID NOT LOCKUP VIRT. ADDR. OF ABORT CORRECTLY
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;"BR 5$" = 000751
044302  042767  160000  133262  8$:     BIC     #160000,SR0     ;CLEAR ERROR BITS IN SR0
044310  077135                          SOB     R1,4$           ;DONE?..NO - GET NEXT DAL & PDR PAIR
044312  000167  000010                  JMP     10$             ;YES...
044316  042767  160000  133246  9$:     BIC     #160000,SR0     ;CLEAR ERROR BITS IN SR0
044324  104000                          EMT                     ;GOT PG. LENGTH ABORT BEFORE IT WAS EXPECTED
                                                                ;FOR TIGHTER SCOPE LOOP
                                                                ;REPLACE ERROR CALL WITH
                                                                ;A "NOP" = 240
```

```
13282
13283  044326  012767  021550  133714  10$:    MOV     #T0250,MMVEC    ;RESTORE NORMAL M.M. TRAP HANDLER
13284                                                                   ;ADDRESS TO M.M. TRAP VECTOR
13285  044334  000167  000064                  JMP     TS404           ;GET TO NEXT TEST
13286
13287                                  ;DAL TABLE FOR UPWARD EXPANSION (NON-ABORT CASES)
13288  044340  100000                  DALTB1: 100000
13289  044342  106100                          106100
13290  044344  102300                          102300
13291  044346  102500                          102500
13292  044350  113700                          113700
13293  044352  104600                          104600
13294  044354  117700                          117700
13295
13296                                  ;PDR TABLE FOR KPDR4 (NON-ABORT CASES)
13297  044356  000006                  PDRTB1: 000006
13298  044360  052006                          052006
13299  044362  045006                          045006
13300  044364  052006                          052006
13301  044366  074406                          074406
13302  044370  025006                          025006
13303  044372  077406                          077406
13304
13305                                  ;DAL TABLE (ABORT CASES)
13306  044374  100100                  DALTB2: 100100
13307  044376  110100                          110100
13308  044400  116600                          116600
13309  044402  112700                          112700
13310  044404  117000                          117000
13311  044406  117700                          117700
13312
13313                                  ;PDR TABLE (ABORT CASES)
13314  044410  000006                  PDRTB2: 000006
13315  044412  030406                          030406
13316  044414  046406                          046406
13317  044416  042006                          042006
13318  044420  073406                          073406
13319  044422  077006                          077006
13320
13332
13333                                  ;*****************************************************************************
  (2)                                  ;TEST 404       PAGE LENGTH FAULTS-DOWNWARD EXPANSION
  (3)                                  ;*****************************************************************************
  (2)  044424                          TS404:
13334  044424  012700  044624          1$:     MOV     #DALTB3,R0      ;DAL TABLE FOR VIRTUAL ADDR'S. TO SELECT PDR4.
13335  044430  012704  044642                  MOV     #PDRTB3,R4      ;PDR TABLE FOR PDR4 (COINCIDES WITH DAL TABLE).
13336  044434  012701  000006                  MOV     #6,R1           ;SET UP LOOP COUNTER.
13337  044440  012767  044602  133602          MOV     #9$,MMVEC       ;SETUP M.M. TRAP VECTOR FOR UNEXPECTED ABORTS
13338  044446  012706  001000                  MOV     #KERSTK,KSP     ;MAKE SURE STACK POINTER IS ALL SET UP
13339
13340                                  ;TEST NON-ABORT CASES (VBA > OR = PLF)
13341  044452  012467  125632          2$:     MOV     (R4)+,KIPDR4    ;LOAD KIPDR4 WITH PAGE LENGTH VALUE
13342  044456  005730                          TST     @(R0)+          ;ACCESS VIRTUAL ADDR. (VBA > OR = PLF)
13343                                                                  ;NO ABORT SHOULD OCCUR!!!
13344  044460  077104                          SOB     R1,2$           ;DONE?...NO- TEST NEXT COMBINATION OF DAL & PDR.
13345
```

```
3346                                        ;TEST ABORT CASES (VBA < PLF)
3347  044462  012701  000005          3$:   MOV   #5,R1          ;SET UP LOOP COUNTER.
3348  044466  012700  044660                MOV   #DALTB4,R0     ;DAL TABLE
3349  044472  012704  044674                MOV   #PDRTB4,R4     ;PDR TABLE
3350  044476  012767  044516  133544        MOV   #6$,@#MMVEC    ;SETUP M.M. TRAP VECTOR FOR EXPECTED ABORT
3351
3352  044504  012467  125600          4$:   MOV   (R4)+,KIPDR4   ;LOAD KIPDR4 WITH PAGE LENGHT VALUE
3353  044510  005730                5$:   TST   @(R0)+         ;ACCESS VIRTUAL ADDR. (VBA < PLF - ABORT TO 6$)
3354  044512  001476                      BEQ   TS405
 (3)  044514  104000                      EMT                  ;EXPECTED PAGE LENGTH ABORT DID NOT OCCUR
3355                                                           ;FOR TIGHTER SCOPE LOOP
3356                                                           ;REPLACE ERROR CALL WITH
3357                                                           ;"BR 5$" = 000776
3358  044516  012706  001000          6$:   MOV   #KERSTK,KSP   ;RESTORE STACK POINTER FOLLOWING ABORT
3359  044522  016767  133044  173034        MOV   SR0,WASSR0     ;READ M.M. STATUS REG. 0
3360  044530  016767  133042  173030        MOV   SR2,WASSR2     ;READ M.M. STATUS REG. 2
3361  044536  012702  040011                MOV   #40011,R2      ;PUT EXPECTED SR0 CONTENTS IN R2
3362  044542  020267  173016                CMP   R2,WASSR0      ;DID SR0 REPORT PG. LENGTH ABORT, PAGE 4, KERNEL?
3363  044546  001401                      BEQ   7$
 (2)  044550  104000                      EMT                  ;SR0 DID NOT REPORT PG. LENGTH ABORT CORRECTLY
3364                                                           ;FOR TIGHTER SCOPE LOOP
3365                                                           ;REPLACE ERROR CALL WITH
3366                                                           ;"BR 5$" = 000757
3367  044552  012703  044510          7$:   MOV   #5$,R3         ;PUT EXPECTED SR2 CONTENTS IN R3
3368  044556  020367  173004                CMP   R3,WASSR2      ;DID SR2 LOCKUP VIRT. ADDR. OF ABORTED INSTRUCTION?
3369  044562  001401                      BEQ   8$
 (2)  044564  104000                      EMT                  ;SR2 DID NOT LOCKUP VIRT. ADDR. OF ABORT CORRECTLY
3370                                                           ;FOR TIGHTER SCOPE LOOP
3371                                                           ;REPLACE ERROR CALL WITH
3372                                                           ;"BR 5$" = 000751
3373  044566  042767  160000  132776  8$:   BIC   #160000,SR0   ;CLEAR ERROR BITS IN SR0
3374  044574  077135                      SOB   R1,4$          ;DONE?..NO - GET NEXT DAL & PDR PAIR
3375  044576  000167  000010                JMP   10$            ;YES...
3376  044602  042767  160000  132762  9$:   BIC   #160000,SR0   ;CLEAR ERROR BITS IN SR0
3377  044610  104000                      EMT                  ;GOT PG. LENGTH ABORT BEFORE IT WAS EXPECTED
3378                                                           ;FOR TIGHTER SCOPE LOOP
3379                                                           ;REPLACE ERROR CALL WITH
3380                                                           ;A "NOP" = 000240
3381
3382  044612  012767  021550  133430  10$:   MOV   #T0250,@#MMVEC  ;RESTORE NORMAL M.M. TRAP HANDLER
3383                                                           ;ADDRESS TO M.M. TRAP VECTOR
3384  044620  000167  000064                JMP   TS405          ;GET TO NEXT TEST
3385
3386                                  ;DAL TABLE FOR DOWNWARD EXPANSION (NON-ABORT CASES)
3387  044624  117700                DALTB3: 117700
3388  044626  111600                        111600
3389  044630  115400                        115400
3390  044632  115200                        115200
3391  044634  104000                        104000
3392  044636  113100                        113100
3393  044640  100000                        100000
3394
3395                                  ;PDR TABLE (NON-ABORT CASES)
3396  044642  077416                PDRTB3: 77416
3397  044644  025416                        25416
3398  044646  032416                        32416
```

```
3399  044650  025416                          25416
3400  044652  003016                          03016
3401  044654  052416                          52416
3402  044656  000016                          00016
3403
3404                                  ;DAL TABLE (ABORT CASES)
3405  044660  117600                  DALTB4: 117600
3406  044662  107600                          107600
3407  044664  101100                          101100
3408  044666  105000                          105000
3409  044670  100700                          100700
3410  044672  100000                          100000
3411
3412                                  ;PDR TABLE (ABORT CASES)
3413  044674  077416                  PDRTB4: 77416
3414  044676  047016                          47016
3415  044700  031016                          31016
3416  044702  035416                          35416
3417  044704  004016                          04016
3418  044706  000416                          00416
3419
3420
3434
3435                                  ;*****************************************************************************************
  (2)                                 ;TEST 405       SR2 BIT TEST
  (3)                                 ;*****************************************************************************************
  (2)  044710                         TS405:
3436  044710  012767  001400  125430  1$:     MOV     #1400,KIPAR3    ;BE SURE PAR3 IS MAPPED TO 24-28K
3437  044716  012767  001400  125424          MOV     #1400,KIPAR4    ;BE SURE PAR4 IS MAPPED TO 24-28K
3438  044724  012767  077406  125354          MOV     #77406,KIPDR3   ;MAP PAGE 3  128 BLOCKS, R/W
3439  044732  012767  077402  125350          MOV     #77402,KIPDR4   ;MAP PAGE 4  128 BLOCKS, READ-ONLY
3440  044740  012700  060002                  MOV     #60002,R0       ;LOAD R0 WITH VIRTUAL ADDR. WHICH USES PDR3
3441  044744  012701  100002                  MOV     #100002,R1      ;LOAD R1 WITH VIRTUAL ADDR. WHICH USES PDR4
3442  044750  012767  044776  133272          MOV     #3$,MMVEC       ;SET M.M. TRAP VECTOR TO 3$
3443  044756  012720  010727          2$:     MOV     #010727,(R0)+   ;LOAD "MOV PC,(PC)+" INSTRUCTION AT ADDR.
3444  044762  005020                  CLR     (R0)+           ;       REACHED THRU PDR/PAR 4.
3445  044764  012720  000137                  MOV     #000137,(R0)+   ;LOAD "JMP @#3$" INSTRUCTION AT VIRT. ADDR.
3446  044770  012710  044776                  MOV     #3$,(R0)        ;       IN CASE R/O VIOL. DOES NOT ABORT
3447  044774  010107                  MOV     R1,PC           ;TRANSFER PROGRAM EXECUTION TO "PAGE 4 INSTRUCTIONS"
3448  044776  012706  001000          3$:     MOV     #KERSTK,KSP     ;RESTORE STACK POINTER
3449  045002  016767  132570  172556          MOV     SR2,WASSR2      ;READ CONTENTS OF STATUS REG 2
3450  045010  020167  172552                  CMP     R1,WASSR2       ;WAS ADDR. OF "RELOCATED - R/O ABORT" LOCKED UP?
3451  045014  001401                  BEQ     4$
  (2)  045016  104000                  EMT                             ;SR2 DID NOT LOCK UP VIRTUAL ADDR. OF R/O VIOL.
3452                                                                  ;FOR TIGHTER SCOPE LOOP
3453                                                                  ;REPLACE ERROR CALL WITH
3454                                                                  ;"BR 2$" = 000757
3455  045020  042767  160000  132544  4$:     BIC     #160000,SR0     ;CLEAR THE ERROR BITS IN SR0
3456  045026  060101                  ADD     R1,R1           ;SETUP TO FORM NEXT VIRTUAL ADDRESS
3457  045030  010100                  MOV     R1,R0           ;SETUP R0 TO FORM NEXT VIRT. ADDR. TO LOAD
3458  045032  052701  100000                  BIS     #100000,R1      ;FORM VIRTUAL ADDR. THAT SHOULD BE LOCKED UP NEXT
3459  045036  052700  060000                  BIS     #60000,R0       ;POINT R0 TO NEXT VIRT. ADDR. TO LOAD
3460  045042  020127  110000                  CMP     R1,#110000      ;HAVE ALL VBA'S 100000-110000 BEEN TESTED?
3461  045046  101743                  BLOS    2$              ;BRANCH IF NO
3462
3463  045050  012767  077406  125232          MOV     #77406,KIPDR4   ;RESTORE PDR4 TO R/W ACCESS
```

```
13464  045056  012767  021550  133164          MOV     #T0250,MMVEC    ;RESTORE ADDRESS OF NORMAL M.M.
13465                                                                   ;TRAP HANDLER TO M.M. VECTOR
13466
13480
13481                                  ;*****************************************************************************
  (2)                                  ;TEST 406       MORE CHECKS OF SR0 & SR2
  (3)                                  ;*****************************************************************************
  (2)  045064                          TS406:
13482  045064  012767  001400  125260  1$:     MOV     #1400,KIPAR5    ;MAP KERNEL PAGE 5 TO 24-28K
13483  045072  012767  000406  125210          MOV     #406,KIPDR4     ;SETUP PDR4 FOR PAGE LENGTH ABORT
13484  045100  012767  077402  125204          MOV     #77402,KIPDR5   ;SETUP PDR5 FOR R/O ABORT
13485  045106  016767  132464  172452  2$:     MOV     SR2,WASSR2      ;READ SR2 TO SEE IF ITS TRACKING
13486  045114  012701  045106                  MOV     #2$,R1          ;PUT EXPECTED VIRTUAL PC IN R1
13487  045120  020167  172442                  CMP     R1,WASSR2       ;DID SR2 CONTAIN VIRTUAL PC AT 2$?
13488  045124  001401                          BEQ     4$
  (2)  045126  104000                          EMT                     ;SR2 NOT TRACKING CORRECTLY
13489                                                                   ;FOR TIGHTER SCOPE LOOP
13490                                                                   ;REPLACE ERROR CALL WITH
13491                                                                   ;"BR 2$" = 000767
13492  045130  016767  132442  172430  4$:     MOV     SR2,WASSR2      ;READ SR2 TO SEE IF ITS TRACKING
13493  045136  012701  045130                  MOV     #4$,R1          ;PUT EXPECTED VIRTUAL PC IN R1
13494  045142  020167  172420                  CMP     R1,WASSR2       ;DID SR2 CONTAIN VIRTUAL PC AT 4$
13495  045146  001401                          BEQ     6$
  (2)  045150  104000                          EMT                     ;SR2 NOT TRACKING CORRECTLY
13496                                                                   ;FOR TIGHTER SCOPE LOOP
13497                                                                   ;REPLACE ERROR CALL WITH
13498                                                                   ;"BR 4$" = 000767
13499  045152  012767  045170  133070  6$:     MOV     #7$,MMVEC       ;PUT ADDRESS OF 7$ IN M.M. TRAP VECTOR
13500  045160  005067  172410                  CLR     $TMP1           ;CLEAR ERROR INDICATOR
13501  045164  005237  100500                  INC     @#100500        ;CAUSE PAGE LENGTH ABORT - TRAP TO 7$
13502  045170  012706  001000          7$:     MOV     #KERSTK,KSP     ;RESTORE STACK POINTER AFTER ABORT
13503  045174  016767  132372  172370          MOV     SR0,$TMP0       ;SAVE SR0'S INFORMATION ON PG. LGTH. ABORT
13504  045202  016767  132370  172366          MOV     SR2,$TMP2       ;SAVE SR2'S INFORMATION ON PG. LGTH. ABORT
13505  045210  012767  045222  133032          MOV     #8$,MMVEC       ;PUT ADDRESS OF 8$ IN M.M. TRAP VECTOR
13506  045216  005237  120000                  INC     @#120000        ;CAUSE R/O ABORT - TRAP TO 8$
13507  045222  012706  001000          8$:     MOV     #KERSTK,KSP     ;RESTORE STACK POINTER AFTER ABORT
13508  045226  016767  132340  172330          MOV     SR0,WASSR0      ;READ SR0 FOLLWOING SECOND KT ABORT
13509  045234  016767  132336  172324          MOV     SR2,WASSR2      ;READ SR2 FOLLOWING SECOND KT ABORT
13510  045242  026767  172324  172314          CMP     $TMP0,WASSR0    ;IS SR0 STILL HOLDING INFO ON FIRST ABORT?
13511  045250  001402                          BEQ     9$              ;BRANCH IF YES
13512  045252  005267  172316                  INC     $TMP1           ;SET ERROR INDICATOR
13513  045256  026767  172314  172302  9$:     CMP     $TMP2,WASSR2    ;DOES SR2 STILL HOLD PC OF FIRST ABORT?
13514  045264  001402                          BEQ     10$             ;BRANCH IF YES
13515  045266  005267  172302                  INC     $TMP1           ;SET ERROR INDICATOR
13516  045272  005767  172276          10$:    TST     $TMP1           ;WERE SR0 OR SR2 CHANGED BY A SECOND ABORT?
13517  045276  001401                          BEQ     11$
  (2)  045300  104000                          EMT                     ;ONE OF STATUS REGS. CHANGED BY SECOND ABORT
13518                                                                   ;FOR TIGHTER SCOPE LOOP
13519                                                                   ;REPLACE ERROR CALL WITH
13520                                                                   ;"BR 6$" = 000726
13521  045302  005067  172266          11$:    CLR     $TMP1           ;CLEAR ERROR INDICATOR
13522  045306  000005                          RESET                   ;EXECUTE A RESET, APPLYING AN "INIT"
13523  045310  005067  132256                  CLR     SR0
13524  045314  016767  132252  172242          MOV     SR0,WASSR0      ;READ SR0
13525  045322  005767  172236                  TST     WASSR0          ;WAS SR0 CLEARED BY THE RESET?
13526  045326  001402                          BEQ     12$             ;BRANCH IF YES
```

```
13527  045330  005267  172240                 INC     $TMP1           ;SRO NOT CLEARED BY A RESET
13528  045334  016767  132236  172224  12$:   MOV     SR2,WASSR2      ;READ SR2
13529  045342  022767  045334  172216         CMP     #12$,WASSR2     ;WAS SR2 UNLOCKED BY A RESET?
13530  045350  001402                         BEQ     13$             ;BRANCH IF YES
13531  045352  005267  172216                 INC     $TMP1           ;SR2 NOT UNLOCKED BY A RESET
13532  045356  005767  172212         13$:    TST     $TMP1           ;WERE SRO & SR2 BOTH "RESET" BY A RESET?
13533  045362  001401                         BEQ     14$
  (2)  045364  104000                         EMT                     ;SRO OR SR2 NOT "RESET" BY A RESET
13534                                                                  ;FOR TIGHTER SCOPE LOOP
13535                                                                  ;REPLACE ERROR CALL WITH
13536                                                                  ;"BR 6$" = 000676
13537  045366  012767  000001  132176  14$:    MOV     #1,SR0          ;TURN MEMORY MANAGEMENT BACK ON
13538  045374  016767  132176  172164  15$:    MOV     SR2,WASSR2      ;READ SR2 TO SEE IF ITS TRACKING AGAIN
13539  045402  012701  045374                 MOV     #15$,R1         ;PUT EXPECTED VIRTUAL PC IN R1
13540  045406  020167  172154                 CMP     R1,WASSR2       ;DID SR2 CONTAIN VIRTUAL PC AT 15$
13541  045412  001401                         BEQ     16$
  (2)  045414  104000                         EMT                     ;SR2 NOT TRACKING CORRECTLY
13542                                                                  ;FOR TIGHTER SCOPE LOOP
13543                                                                  ;REPLACE ERROR CALL WITH
13544                                                                  ;"BR 6$" = 000663
13545  045416  012767  077406  124664  16$:    MOV     #77406,KIPDR4   ;RESET PDR4 TO 128 BLKS, R/W
13546  045424  012767  077406  124660         MOV     #77406,KIPDR5   ;RESET PDR5 TO 128 BLKS, R/W
13547  045432  012767  021550  132610         MOV     #T0250,MMVEC    ;RESTORE ADDRESS OF NORMAL MEMORY
13548                                                                  ;MANAGEMENT TRAP ROUTINE TO M.M. VECTOR
13549
13561
13562                          ;*****************************************************************************
  (2)                          ;TEST 407       USER ABORT PICKS UP KERNEL SPACE VECTOR
  (3)                          ;*****************************************************************************
  (2)  045440                  TS407:
13563  045440  004767  002614         1$:     JSR     PC,TOFF         ;TURN OFF T-BIT TRAPPING FOR THIS TEST
13564  045444  005067  132326         2$:     CLR     PSW             ;GO TO KERNEL MODE
13565  045450  012706  001000                 MOV     #KERSTK,KSP     ;SETUP KERNEL STACK PTR.
13566  045454  012767  001400  132156         MOV     #1400,UIPAR0    ;MAP USER PAGE 0 TO 24K
13567  045462  012737  045532  000004         MOV     #4$,@#4         ;LOAD KERNEL VECTOR 4 (LOC.4) WITH 4$
13568  045470  012737  000340  000006         MOV     #340,@#6        ;LOAD VECTOR+2 WITH NEW PSW
13569  045476  012767  140000  132272         MOV     #140000,PSW     ;GO TO USER MODE
13570  045504  012706  000600                 MOV     #USESTK,USP     ;SETUP USER STACK PTR.
13571  045510  012737  045530  000004         MOV     #3$,@#4         ;LOAD USER VECTOR 4 (LOC. 60004) WITH 3$
13572  045516  012737  000340  000006         MOV     #340,@#6        ;LOAD VECTOR+2 WITH NEW PSW
13573  045524  005767  112250                 TST     160000          ;CAUSE TIMEOUT ERROR TRAP TO "4"
13574                                                                  ;SHOULD PICK UP NEW PC=4$ FROM KERNEL
13575                                                                  ;LOC. 4, NOT PC=3$ FROM USER LOC. 4 (=60004)
13576  045530                  3$:
13577  045530  104000                         EMT                     ;DID NOT TRAP THRU KERNEL SPACE
13578                                                                  ;FOR TIGHTER SCOPE LOOP
13579                                                                  ;REPLACE ERROR CALL WITH
13580                                                                  ;"BR 2$" = 000740
13581  045532  005067  132240         4$:     CLR     PSW             ;BE SURE BACK IN KERNEL MODE
13582  045536  012706  001000                 MOV     #KERSTK,KSP     ;RESTORE KERNEL S.P. IN CASE IT CHANGED
13583  045542  005067  132072                 CLR     UIPAR0          ;REMAP USER PAGE 0 TO 0-4K
13584  045546  012767  140000  132222         MOV     #140000,PSW     ;GO TO USER MODE
13585  045554  012706  000600                 MOV     #USESTK,USP     ;RESTORE USER STACK POINTER
13586  045560  005067  132212                 CLR     PSW             ;GO BACK TO KERNEL MODE
13587  045564  012737  021526  000004         MOV     #T04,@#4        ;RESTORE ADDR. OF NORMAL CPU TRAP HANDLER TO 4
13588  045572  004767  002516                 JSR     PC,TON          ;TURN T-BIT TRAPPING BACK ON
```

```
                                        ;*******************************************************************************
                                        ;TEST 410       RTI IN USER MODE DOES NOT CHANGE PSW
                                        ;*******************************************************************************
045576                                  TS410:

045576  012702  170000                          MOV     #170000,R2      ;LOAD "PRESENT & EXPECTED" PSW VALUE INTO R2
045602  010267  132170                  2$:     MOV     R2,PSW          ;GO TO USER MODE-PRIORITY 0
045606  012746  000340                          MOV     #340,-(SP)      ;PUT A NEW PSW (PRIORITY=7) ON STACK
045612  012746  045620                          MOV     #3$,-(SP)       ;PUT NEW PC ON THE STACK
045616  000002                                  RTI                     ;DO AN RTI FROM USER MODE
045620  016701  132152                  3$:     MOV     PSW,R1          ;READ NEW PSW INTO R1
045624  042701  007437                          BIC     #7437,R1        ;MASK OFF COND. CODE, T-BIT, AND UNUSED BITS
045630  005067  132142                          CLR     PSW             ;GO BACK TO KERNEL MODE
045634  020201                                  CMP     R2,R1           ;DID PSW STAY IN USER, PRIORITY=0?
045636  001401                                  BEQ     TS411
045640  104000                                  EMT                     ;PSW CHANGED BY AN RTI FROM USER
                                                                        ;FOR A TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
                                                                        ;"BR=2$" = 000760


                                        ;*******************************************************************************
                                        ;TEST 411       KT ERROR SERVICED BEFORE TIMEOUT ERROR
                                        ;*******************************************************************************
045642                                  TS411:
045642  012705  077006                  1$:     MOV     #77006,R5       ;LOAD PDR7 DATA INTO R5
045646  010567  124444                          MOV     R5,KIPDR7       ;MAP PAGE 7 R/W  PLF=176
045652  012737  045672  000004                  MOV     #3$,@#4         ;SET CPU TRAP VECTOR TO ADDRESS OF 3$
045660  012737  045674  000250                  MOV     #4$,@#250       ;SET M.M. TRAP VECTOR TO ADDRESS OF 4$
045666  005237  177700                  2$:     INC     @#177700        ;CAUSE PLF ABORT AND POTENTIAL TIMEOUT
045672                                  3$:
045672  104000                                  EMT                     ;TRAPPED THRU CPU TRAP VECTOR BUT SHOULDN'T HAVE
                                                                        ;FOR TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
                                                                        ;"BR 2$" = 000776
045674  012706  001000                  4$:     MOV     #KERSTK,KSP     ;RESTORE STACK POINTER AFTER TRAPPING
045700  016767  131666  171656                  MOV     SR0,WASSR0      ;READ STATUS REG.0
045706  016767  131664  171652          5$:     MOV     SR2,WASSR2      ;READ STATUS REG. 2
045714  012700  040017                          MOV     #40017,R0       ;LOAD EXPECTED SR0 CONTENTS INTO R0
045720  020067  171640                          CMP     R0,WASSR0       ;SR0 PLF ERROR BIT SET?
045724  001401                                  BEQ     6$
045726  104000                                  EMT                     ;SR0 DIDN'T REPORT PLF ERROR
                                                                        ;FOR TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
                                                                        ;"BR 2$" = 000741
045730  012701  045666                  6$:     MOV     #2$,R1          ;LOAD EXPECTED SR2 CONTENTS INTO R1
045734  020167  171626                          CMP     R1,WASSR2       ;WAS SR2 LOCKED BY PLF ABORT?
045740  001401                                  BEQ     7$
045742  104000                                  EMT                     ;SR2 DIDN'T LOCK UP VIRTUAL ADDRESS
                                                                        ;FOR TIGHTER SCOPE LOOP
                                                                        ;REPLACE ERROR CALL WITH
                                                                        ;"BR 2$" = 000741
045744  042767  160000  131620          7$:     BIC     #160000,SR0     ;CLEAR ERROR BITS THAT WERE SET IN SR0
045752  012737  021526  000004                  MOV     #T04,@#4        ;RESTORE ADDRESS OF NORMAL CPU TRAP HANDLER
045760  012737  021550  000250                  MOV     #T0250,@#250    ;RESTORE ADDRESS OF NORMAL M.M. TRAP HANDLER
```

```
13653  045766  012767  077406  124322          MOV     #77406,KIPDR7   ;REMAP PAGE 7 TO READ/WRITE PLF=177
13654
13669
13670                                  ;*****************************************************************************
  (2)                                  ;TEST 412       PC & PSW SAVED FOR KT ERROR DURING SERVICE OF TIMEOUT ERROR
  (3)                                  ;*****************************************************************************
  (2)  045774                          TS412:
13671  045774  004767  002260          1$:     JSR     PC,TOFF         ;TURN T-BIT TRAPPING OFF FOR THIS TEST
13672  046000  012767  001400  131640          MOV     #1400,UIPAR3    ;MAP USER PAGE 3 TO 24-28K
13673  046006  012767  001400  131634          MOV     #1400,UIPAR4    ;MAP USER PAGE 4 TO 24-28K
13674  046014  012767  077402  131564          MOV     #77402,UIPDR3   ;MAP USER PAGE 3 READ-ONLY
13675  046022  012767  077406  131560          MOV     #77406,UIPDR4   ;MAP USER PAGE 4 READ/WRITE
13676  046030  012737  046076  000004          MOV     #4$,@#4         ;LOAD ADDRESS OF 4$ IN CPU (TIMEOUT) VECTOR
13677  046036  012737  140017  000006          MOV     #140017,@#6     ;LOAD PSW THAT SHOULD BE PUT ON STACK IN VECTOR+2
13678  046044  012737  046076  000250          MOV     #4$,@#250       ;LOAD ADDRESS OF 4$ IN M.M. TRAP VECTOR
13679  046052  012737  000340  000252          MOV     #340,@#252      ;LOAD A KERNEL PSW IN MMVEC+2
13680  046060  012767  140000  131710  2$:     MOV     #140000,PSW     ;GO TO USER MODE
13681  046066  012706  100002                  MOV     #100002,USP     ;SET USER STACK PTR. SO SECOND PUSH IS IN PG. 3
13682  046072  005737  177700          3$:     TST     @#177700        ;CAUSE TIMEOUT ERROR THAT WILL CAUSE
13683                                                                  ;R/O ERROR WHEN TRY TO SAVE OLD PC
13684  046076  016601  000002          4$:     MOV     2(KSP),R1       ;PUT PSW SAVED ON KERNEL STACK INTO R1
13685  046102  011603                          MOV     (KSP),R3        ;PUT PC SAVED ON KERNEL STACK INTO R3
13686  046104  016767  131462  171452          MOV     SR0,WASSR0      ;READ THE CONTENTS OF M.M. STATUS REG. 0
13687  046112  016767  131460  171446          MOV     SR2,WASSR2      ;READ THE CONTENTS OF M.M. STATUS REG. 2
13688  046120  042767  160000  131444          BIC     #160000,SR0     ;CLEAR THE ERROR BITS IN SR0
13689  046126  005067  131644                  CLR     PSW             ;BE SURE IN KERNEL MODE
13690  046132  012706  001000                  MOV     #KERSTK,KSP     ;RESTORE KERNEL STACK POINTER
13691  046136  012767  140000  131632          MOV     #140000,PSW     ;GO TO USER MODE
13692  046144  012706  000600                  MOV     #USESTK,USP     ;RESTORE USER STACK POINTER
13693  046150  005067  131622                  CLR     PSW             ;GO BACK TO KERNEL MODE
13694  046154  005067  171412                  CLR     $TMP0           ;CLEAR ERROR INDICATOR
13695  046160  020127  170017                  CMP     R1,#170017      ;WAS THE PSW SAVED THE ONE PICKED UP BY THE
13696                                                                  ;TIMEOUT TRAP FROM ERRVEC+2?
13697                                                                  ;VALUE 170017 = PSW FROM LOC. 6 WITH
13698                                                                  ;PREVIOUS MODE BITS = USER
13699  046164  001402                          BEQ     5$              ;BRANCH IF YES
13700  046166  005267  171400                  INC     $TMP0           ;WRONG PSW SAVED DURING "DOUBLE ERROR" SEQUENCE
13701  046172  020327  046076          5$:     CMP     R3,#3$+4        ;WAS THE PC AT THE TIME OF THE TIMEOUT ERROR
13702                                                                  ;SAVED ON THE STACK?
13703  046176  001402                          BEQ     6$              ;BRANCH IF YES
13704  046200  005267  171366                  INC     $TMP0           ;WRONG PC SAVED DURING TRAP SEQUENCE
13705  046204  026727  171354  020147  6$:     CMP     WASSR0,#20147   ;DID SR0 REPORT - USER, PAGE 3, R/O ABORT?
13706  046212  001402                          BEQ     7$              ;BRANCH IF YES
13707  046214  005267  171352                  INC     $TMP0           ;SR0 DID NOT REPORT R/O ABORT
13708  046220  026727  171342  046072  7$:     CMP     WASSR2,#3$      ;DID SR2 LOCK UP VIRTUAL ADDR. OF LAST
13709                                                                  ;INSTRUCTION SUCCESSFULLY FETCHED?
13710  046226  001402                          BEQ     8$              ;BRANCH IF YES
13711  046230  005267  171336                  INC     $TMP0           ;SR2 DID NOT LOCK UP ADDR. OF TIMEOUT INST.
13712  046234  005767  171332          8$:     TST     $TMP0           ;ANY "ERRORS" DURING TRAP SEQUENCE?
13713  046240  001401                          BEQ     9$
  (2)  046242  104000                          EMT                     ;THE WRONG PC OR PSW WERE SAVED
13714                                                                  ;OR SR0 OR SR2 DID NOT REPORT R/O
13715                                                                  ;ERROR DURING TIMEOUT - KT TRAP
13716                                                                  ;SEQUENCE
13717                                                                  ;FOR TIGHTER SCOPE LOOP
13718                                                                  ;REPLACE ERROR CALL WITH
```

```
13719                                                   ;"BR 2$" = 000710
13720  046244  012737  021526  000004  9$:     MOV     #T04,@#4        ;RESTORE ADDRESS OF NORMAL CPU TRAP HANDLER
13721  046252  012737  000340  000006          MOV     #340,@#6        ;RELOAD ERRVEC+2 WITH KERNEL PSW
13722  046260  012737  021550  000250          MOV     #T0250,@#250    ;RESTORE ADDRESS OF NORMAL M.M. TRAP HANDLER
13723  046266  012767  077406  131312          MOV     #77406,UIPDR3   ;REMAP USER PAGE 3 READ/WRITE
13724  046274  004767  002014                  JSR     PC,TON          ;TURN T-BIT TRAPPING BACK ON
13725                                          ;*************************************************************************
13726                                          ;*
13727                                          ;*      THIS GROUP OF TESTS WILL TEST ALL THE LOGIC ASSOCIATED WITH
13728                                          ;*      THE "MOVE FROM PREVIOUS" AND MOVE TO PREVIOUS" INSTRUCTIONS.
13729                                          ;*
13730                                          ;*************************************************************************
13731
13732
13744                                          ;*************************************************************************
  (2)                                         ;TEST 413       MOVE FROM PREVIOUS (USER) I-SPACE
  (3)                                         ;*************************************************************************
  (2)  046300
13745  046300  005067  124034          TS413:
13746  046304  012767  000200  124030  1$:     CLR     KIPAR0          ;MAP KERNEL PAGE 0 TO 0-4K
13747  046312  012767  000400  124024          MOV     #200,KIPAR1     ;MAP KERNEL PAGE 1 TO 4-8K
13748  046320  012767  000600  124020          MOV     #400,KIPAR2     ;MAP KERNEL PAGE 2 TO 8-12K
13749  046326  012767  001400  124014          MOV     #600,KIPAR3     ;MAP KERNEL PAGE 3 TO 12-16K
13750  046334  012767  007600  124014          MOV     #1400,KIPAR4    ;MAP KERNEL PAGE 4 TO 24-28K
13751  046342  012700  077406                  MOV     #7600,KIPAR7    ;MAP KERNEL PAGE 7 TO THE I/O PAGE
13752                                          MOV     #77406,R0       ;MAKE ALL KERNEL I-SPACE PAGES RESIDENT
13753  046346  012702  000010                                          ;READ/WRITE, LENGTH 200 BLOCKS
13754  046352  012701  172300                  MOV     #10,R2          ;SET LOOP COUNTER TO 8
13755  046356  010021                  2$:     MOV     #KIPDR0,R1      ;PUT ADDRESS OF FIRST PDR IN R1
13756  046360  077202                          MOV     R0,(R1)+        ;LOAD PDR WITH 77406
13757  046362  012702  000010                  SOB     R2,2$           ;LOOP TO 2$ UNTIL ALL PDRS LOADED
13758  046366  012701  177600                  MOV     #10,R2          ;SET LOOP COUNTER TO 8
13759  046372  010021                  3$:     MOV     #UIPDR0,R1      ;PUT ADDRESS OF FIRST PDR IN R1
13760  046374  077202                          MOV     R0,(R1)+        ;LOAD PDR WITH 77406
13761  046376  012767  000000  131234          SOB     R2,3$           ;LOOP TO 3$ UNTIL ALL PDRS LOADED
13762  046404  012767  000200  131230          MOV     #000,UIPAR0     ;MAP USER I PAGE 0 TO 0-4K
13763  046412  012767  000400  131224          MOV     #200,UIPAR1     ;MAP USER I PAGE 1 TO 4-8K
13764  046420  012767  000600  131220          MOV     #400,UIPAR2     ;MAP USER I PAGE 2 TO 8-12K
13765  046426  012767  007600  131222          MOV     #600,UIPAR3     ;MAP USER I PAGE 3 TO 12-16K
13766  046434                          4$:     MOV     #7600,UIPAR7    ;MAP USER I PAGE 7 TO THE I/O PAGE
13767  046434  012767  077406  123646          MOV     #77406,KIPDR4   ;KERNEL I-SPACE PAGE 4 READ/WRITE
13768  046442  012767  001400  123700          MOV     #1400,KIPAR4    ;MAP KERNEL I PAGE 4 TO 24K
13769  046450  012767  001400  131172          MOV     #1400,UIPAR4    ;MAP USER I PAGE 4 TO 24K
13770  046456  012700  036514                  MOV     #36514,R0       ;LOAD DATA PATTERN INTO R0
13771  046462  010037  100000                  MOV     R0,@#100000     ;LOAD DATA PATTERN INTO PHY 140000
13772  046466  012767  046764  131554          MOV     #23$,MMVEC      ;SET M.M. VECTOR TO 23$
13773  046474  105067  123610                  CLRB    KIPDR4          ;MAKE KERNEL I-SPACE PAGE 4 NON-RESIDENT
13774                                          ;THE FOLLOWING WILL TEST DSTM=0 MFPI
13775                                          ;
13776  046500  012767  030340  131270  5$:     MOV     #030340,PSW     ;MAKE PREVIOUS MODE USER
13777  046506  006506                  6$:     MFPI    USP             ;PUT USER STACK POINTER ON KERNEL
13778                                                                  ;STACK
13779  046510  022706  001000                  CMP     #KERSTK,KSP     ;WAS SOMETHING PUSHED ON STACK AT 6$
13780  046514  001405                          BEQ     7$              ;BRANCH IF NOTHING WAS PUSHED
13781  046516  012600                          MOV     (KSP)+,R0       ;POP KERNEL STACK INTO R0
13782  046520  012701  000600                  MOV     #USESTK,R1      ;EXPECTING TO GET 700 AS USP
```

```
13783  046524  020001                          CMP     R0,R1           ;DID YOU GET THE RIGHT POINTER?
13784  046526  001401                          BEQ     8$
  (1)  046530                          7$:
  (2)  046530  104000                          EMT                     ;WRONG THING WAS PUSHED ON STACK
13785                                                                  ;FOR TIGHTER SCOPE LOOP
13786                                                                  ;REPLACE ERROR CALL WITH
13787                                                                  ;"BR 5$" = 000763
13788  046532                          8$:     ;THE FOLLOWING WILL TEST DSTM=1 MFPI.
13789  046532  012700  036514                  MOV     #36514,R0       ;RELOAD DATA PATTERN IN R0
13790  046536  012767  030340  131232  9$:     MOV     #030340,PSW     ;MAKE PREVIOUS MODE USER
13791  046544  012702  100000                  MOV     #100000,R2      ;LOAD VIRTUAL ADDRESS INTO R2
13792  046550  006512                          MFPI    (R2)            ;READ FROM PHYSICAL 140000
13793  046552  012601                          MOV     (KSP)+,R1       ;POP KERNEL STACK INTO R1
13794  046554  020001                          CMP     R0,R1           ;WAS DATA FETCHED SAME AS STORED
13795  046556  001401                          BEQ     10$
  (2)  046560  104000                          EMT                     ;WRONG DATA WAS FETCHED
13796                                                                  ;FOR TIGHTER SCOPE LOOP
13797                                                                  ;REPLACE ERROR CALL WITH
13798                                                                  ;"BR 9$" = 000766
13799  046562                          10$:    ;THE FOLLOWING WILL TEST DSTM=2 MFPI.
13800  046562  012767  030340  131206  11$:    MOV     #030340,PSW     ;MAKE PREVIOUS MODE USER
13801  046570  012702  100000                  MOV     #100000,R2      ;LOAD VIRTUAL ADDRESS INTO R2
13802  046574  006522                          MFPI    (R2)+           ;READ FROM PHYSICAL 140000
13803  046576  012601                          MOV     (KSP)+,R1       ;POP KERNEL STACK INTO R1
13804  046600  020001                          CMP     R0,R1           ;WAS DATA FETCHED SAME AS STORED
13805  046602  001401                          BEQ     12$
  (2)  046604  104000                          EMT                     ;WRONG DATA WAS FETCHED
13806                                                                  ;FOR TIGHTER SCOPE LOOP
13807                                                                  ;REPLACE ERROR CALL WITH
13808                                                                  ;"BR 11$" = 000766
13809  046606                          12$:    ;THE FOLLOWING WILL TEST DSTM=3 MFPI.
13810  046606  012767  030340  131162  13$:    MOV     #030340,PSW     ;MAKE PREVIOUS MODE USER
13811  046614  006537  100000                  MFPI    @#100000        ;READ FROM PHYSICAL 140000
13812  046620  012601                          MOV     (KSP)+,R1       ;POP KERNEL STACK INTO R1
13813  046622  020001                          CMP     R0,R1           ;WAS DATA FETCHED SAME AS STORED
13814  046624  001401                          BEQ     14$
  (2)  046626  104000                          EMT                     ;WRONG DATA WAS FETCHED
13815                                                                  ;FOR TIGHTER SCOPE LOOP
13816                                                                  ;REPLACE ERROR CALL WITH
13817                                                                  ;"BR 13$" = 000767
13818  046630                          14$:    ;THE FOLLOWING WILL TEST DSTM=4 MFPI.
13819  046630  012767  030340  131140  15$:    MOV     #030340,PSW     ;MAKE PREVIOUS MODE USER
13820  046636  012702  100002                  MOV     #100002,R2      ;LOAD VIRTUAL ADDRESS INTO R2
13821  046642  006542                          MFPI    -(R2)           ;READ FROM PHYSICAL 140000
13822  046644  012601                          MOV     (KSP)+,R1       ;POP KERNEL STACK INTO R1
13823  046646  020001                          CMP     R0,R1           ;WAS DATA FETCHED SAME AS STORED
13824  046650  001401                          BEQ     16$
  (2)  046652  104000                          EMT                     ;WRONG DATA WAS FETCHED
13825                                                                  ;FOR TIGHTER SCOPE LOOP
13826                                                                  ;REPLACE ERROR CALL WITH
13827                                                                  ;"BR 15$" = 000766
13828  046654                          16$:
13829                                  ;THE FOLLOWING WILL TEST DSTM=5 MFPI.
13830                                  ;
13831  046654  012767  030340  131114  17$:    MOV     #030340,PSW     ;MAKE PREVIOUS MODE USER
13832  046662  012767  100000  170706          MOV     #100000,$TMP2   ;LOAD TEST LOC. VIRT. ADDR INTO LOC. $TMP2
```

```
13833  046670  012702  037600              MOV     #<$TMP2+2>,R2   ;LOAD ADDR. OF $TMP2+2 INTO R2
13834  046674  006552                      MFPI    @-(R2)          ;READ FROM PHYSICAL 140000
13835  046676  012601                      MOV     (KSP)+,R1       ;POP KERNEL STACK INTO R1
13836  046700  020001                      CMP     R0,R1           ;WAS DATA FETCHED SAME AS STORED
13837  046702  001401                      BEQ     18$
  (2)  046704  104000                      EMT                     ;WRONG DATA WAS FETCHED
13838                                                               ;FOR TIGHTER SCOPE LOOP
13839                                                               ;REPLACE ERROR CALL WITH
13840                                                               ;"BR 17$" = 000763
13841  046706                      18$:    ;THE FOLLOWING WILL TEST DSTM=6 MFPI.
13842                                      ;
13843  046706  012767  030340  131062 19$:  MOV     #030340,PSW     ;MAKE PREVIOUS MODE USER
13844  046714  005002                      CLR     R2              ;MAKE REGISTER 2 A ZERO
13845  046716  006562  100000              MFPI    100000(R2)      ;READ FROM PHYSICAL 140000
13846  046722  012601                      MOV     (KSP)+,R1       ;POP KERNEL STACK INTO R1
13847  046724  020001                      CMP     R0,R1           ;WAS DATA FETCHED SAME AS STORED
13848  046726  001401                      BEQ     20$
  (2)  046730  104000                      EMT                     ;WRONG DATA WAS FETCHED
13849                                                               ;FOR TIGHTER SCOPE LOOP
13850                                                               ;REPLACE ERROR CALL WITH
13851                                                               ;"BR 19$" = 000766
13852  046732                      20$:    ;THE FOLLOWING WILL TEST DSTM=7 MFPI.
13853                                      ;
13854  046732  012767  030340  131036 21$:  MOV     #030340,PSW     ;MAKE PREVIOUS MODE USER
13855  046740  012767  100000  170630      MOV     #100000,$TMP2   ;LOAD TEST LOC. V.A. INTO $TMP2
13856  046746  012702  037576              MOV     #$TMP2,R2       ;LOAD ADDRESS OF $TMP2 INTO R2
13857  046752  006572  000000              MFPI    @0(R2)          ;USE $TMP2 TO FETCH VIRTUAL
13858                                                               ;ADDRESS OF 140000
13859  046756  012601                      MOV     (KSP)+,R1       ;POP KERNEL STACK INTO R1
13860  046760  020001                      CMP     R0,R1           ;WAS DATA FETCHED SAME AS STORED
13861  046762  001401                      BEQ     22$
  (1)  046764                      23$:
  (2)  046764  104000                      EMT                     ;WRONG DATA WAS FETCHED
13862                                                               ;FOR TIGHTER SCOPE LOOP
```

```
13864                                                           ;REPLACE ERROR CALL WITH
13865                                                           ;"BR 21$" = 000762
13866  046766  012767  021550  131254  22$:   MOV     #T0250,MMVEC    ;SET M.M. VECTOR TO NORMAL ROUTINE
13867
13868
```

```
                                        ;*******************************************************************************
                                        ;TEST 414        MOVE TO PREVIOUS (USER) I-SPACE
                                        ;*******************************************************************************
  046774                                TS414:
  046774  012767  077406  123306  1$:     MOV     #77406,KIPDR4    ;KERNEL I-SPACE PAGE 4 READ/WRITE
  047002  012767  077406  130600          MOV     #77406,UIPDR4    ;USER I-SPACE PAGE 4 READ/WRITE
  047010  012767  001400  123332          MOV     #1400,KIPAR4     ;MAP KERNEL I PAGE 4 TO 24K
  047016  012767  001400  130624          MOV     #1400,UIPAR4     ;MAP USER I PAGE 4 TO 24K
  047024  012767  047516  131216          MOV     #20$,MMVEC       ;SET M.M. VECTOR TO 20$
                                          ;THE FOLLOWING WILL TEST DSTM=0 MTPI
                                          ;
  047032  012767  030340  130736  2$:     MOV     #030340,PSW      ;MAKE PREVIOUS MODE USER
  047040  012746  007777                  MOV     #7777,-(KSP)     ;PUSH DATA ON KERNEL STACK
  047044  006606                          MTPI    USP              ;LOAD USER STACK POINTER
  047046  006506                          MFPI    USP              ;READ USER STACK POINTER
  047050  012601                          MOV     (KSP)+,R1        ;POP KERNEL STACK INTO R1
  047052  022701  007777                  CMP     #7777,R1         ;WAS USER STACK POINTER CHANGED
  047056  001401                          BEQ     3$
  047060  104000                          EMT                      ;USER STACK POINTER NOT CHANGED
                                                                   ;FOR TIGHTER SCOPE LOOP
                                                                   ;REPLACE ERROR CALL WITH
                                                                   ;"BR 2$" = 000764
  047062  012767  030340  130706  3$:     MOV     #030340,PSW      ;MAKE PREVIOUS MODE USER
  047070  012746  000600                  MOV     #USESTK,-(KSP)   ;GET READY TO RESTORE USER S. POINT
  047074  006606                          MTPI    USP              ;RESTORE USER STACK POINTER
  047076                          4$:     ;THIS WILL TEST DSTM = 1 MTPI.
  047076  012702  100000                  MOV     #100000,R2       ;LOAD VIRTUAL ADDRESS INTO R2
  047102  012700  125252                  MOV     #125252,R0       ;LOAD TEST DATA INTO R0
  047106  010046                  5$:     MOV     R0,-(KSP)        ;PUSH TEST DATA ON KERNEL STACK
  047110  105067  123174                  CLRB    KIPDR4           ;MAKE KERNEL I PAGE 4 NON-RESIDENT
  047114  006612                          MTPI    (R2)             ;LOAD TEST DATA INTO PHYSICAL 140000
  047116  112767  000006  123164          MOVB    #006,KIPDR4      ;MAKE KERNEL PAGE 4 RESIDENT
  047124  011201                          MOV     (R2),R1          ;READ FROM ADDRESS 140000
  047126  020001                          CMP     R0,R1            ;SEE IF DATA WAS STORED AT CORRECT PLACE
  047130  001401                          BEQ     6$
  047132  104000                          EMT                      ;INCORRECT STORE
                                                                   ;FOR TIGHTER SCOPE LOOP
                                                                   ;REPLACE ERROR CALL WITH
                                                                   ;"BR 5$" = 000765
  047134                          6$:     ;THE FOLLOWING WILL TEST DSTM=2 MTPI.
                                          ;
  047134  012767  030340  130634          MOV     #030340,PSW      ;MAKE PREVIOUS MODE USER
  047142  012700  125252                  MOV     #125252,R0       ;LOAD TEST DATA INTO R0
  047146  012702  100000                  MOV     #100000,R2       ;LOAD VIRTUAL ADDRESS INTO R2
  047152  010046                  8$:     MOV     R0,-(KSP)        ;PUSH TEST DATA ON KERNEL STACK
  047154  105067  123130                  CLRB    KIPDR4           ;MAKE KERNEL PAGE 4 NON-RESIDENT
  047160  006612                          MTPI    (R2)             ;LOAD TEST DATA INTO PHYSICAL 140000
  047162  112767  000006  123120          MOVB    #006,KIPDR4      ;MAKE KERNEL PAGE 4 RESISENT
  047170  013701  100000                  MOV     @#100000,R1      ;READ FROM ADDRESS 140000
  047174  020001                          CMP     R0,R1            ;SEE IF DATA WAS STORED CORRECTLY
  047176  001401                          BEQ     9$
  047200  104000                          EMT                      ;INCORRECT STORE
                                                                   ;FOR TIGHTER SCOPE LOOP
                                                                   ;REPLACE ERROR CALL WITH
                                                                   ;"BR 8$" = 000764
  047202                          9$:     ;THIS WILL TEST DSTM = 3 MTPI.
```

```
13931  047202  012767  030340  130566          MOV     #030340,PSW     ;MAKE PREVIOUS MODE USER
13932  047210  012700  052525                  MOV     #52525,R0       ;LOAD TEST DATA INTO R0
13933  047214  010046                  10$:    MOV     R0,-(KSP)       ;PUSH TEST DATA ON KERNEL STACK
13934  047216  105067  123066                  CLRB    KIPDR4          ;MAKE KERNEL I PAGE 4 NON-RESIDENT
13935  047222  006637  100000                  MTPI    @#100000        ;LOAD TEST DATA INTO PHYSICAL 140000
13936  047226  112767  000006  123054          MOVB    #006,KIPDR4     ;MAKE KERNEL PAGE 4 RESIDENT
13937  047234  013701  100000                  MOV     @#100000,R1     ;READ FROM ADDRESS 140000
13938  047240  020001                          CMP     R0,R1           ;SEE IF DATA WAS STORED CORRECTLY
13939  047242  001531                          BEQ     TS415
  (3)  047244  104000                          EMT                     ;
13940  047246  001401                          BEQ     11$
  (2)  047250  104000                          EMT                     ;INCORRECT STORE
13941                                                                  ;FOR TIGHTER SCOPE LOOP
13942                                                                  ;REPLACE ERROR CALL WITH
13943                                                                  ;"BR 10$" = 000763
13944  047252                  11$:    ;THIS WILL TEST DSTM = 4 MTPI.
13945  047252  012767  030340  130516          MOV     #030340,PSW     ;MAKE PREVIOUS MODE USER
13946  047260  012700  125252                  MOV     #125252,R0      ;LOAD TEST DATA INTO R0
13947  047264  010046                  12$:    MOV     R0,-(KSP)       ;PUSH TEST DATA ON KERNEL STACK
13948  047266  012702  100002                  MOV     #100002,R2      ;LOAD VIRTUAL ADDRESS INTO R2
13949  047272  105067  123012                  CLRB    KIPDR4          ;MAKE KERNEL I PAGE 4 NON-RESIDENT
13950  047276  006642                          MTPI    -(R2)           ;LOAD TEST DATA INTO PHYSICAL 140000
13951  047300  112767  000006  123002          MOVB    #006,KIPDR4     ;MAKE KERNEL PAGE 4 RESIDENT
13952  047306  013701  100000                  MOV     @#100000,R1     ;READ FROM ADDRESS 140000
13953  047312  020001                          CMP     R0,R1           ;SEE IF DATA WAS STORED CORRECTLY
13954  047314  001401                          BEQ     13$
  (2)  047316  104000                          EMT                     ;INCORRECT STORE
13955                                                                  ;FOR TIGHTER SCOPE LOOP
13956                                                                  ;REPLACE ERROR CALL WITH
13957                                                                  ;"BR 12$" = 000762
13958  047320                  13$:    ;THE FOLLOWING WILL TEST DSTM=5 MTPI.
13959                                  ;
13960  047320  012767  030340  130450          MOV     #030340,PSW     ;MAKE PREVIOUS MODE USER
13961  047326  012700  052525                  MOV     #52525,R0       ;LOAD TEST DATA INTO R0
13962  047332  012702  037600                  MOV     #<$TMP2+2>,R2   ;LOAD ADDR. OF LOC. $TMP2+2 INTO R2
13963  047336  012767  100000  170232          MOV     #100000,$TMP2   ;LOAD VIRT. ADDR. OF TEST LOC. INTO $TMP2
13964  047344  010046                  14$:    MOV     R0,-(KSP)       ;PUSH TEST DATA ON KERNEL STACK
13965  047346  105067  122736                  CLRB    KIPDR4          ;MAKE KERNEL PAGE 4 NON-RESIDENT
13966  047352  006652                          MTPI    @-(R2)          ;LOAD TEST DATA INTO PHYSICAL 140000
13967  047354  112767  000006  122726          MOVB    #006,KIPDR4     ;MAKE KERNEL PAGE 4 RESIDENT
13968  047362  013701  100000                  MOV     @#100000,R1     ;READ FROM ADDRESS 140000
13969  047366  020001                          CMP     R0,R1           ;SEE IF DATA WAS STORED CORRECTLY
13970  047370  001401                          BEQ     15$
  (2)  047372  104000                          EMT                     ;INCORRECT STORE
13971                                                                  ;FOR TIGHTER SCOPE LOOP
13972                                                                  ;REPLACE ERROR CALL WITH
13973                                                                  ;"BR 14$" = 000764
13974  047374                  15$:    ;THIS WILL TEST DSTM = 6 MTPI.
13975                                  ;
13976  047374  012767  030340  130374          MOV     #030340,PSW     ;MAKE PREVIOUS MODE USER
13977  047402  012700  052525                  MOV     #52525,R0       ;LOAD TEST DATA INTO R0
13978  047406  005002                          CLR     R2              ;MAKE REGISTER 2 ZERO
13979  047410  010046                  16$:    MOV     R0,-(KSP)       ;PUSH TEST DATA ON KERNEL STACK
13980  047412  105067  122672                  CLRB    KIPDR4          ;MAKE KERNEL I PAGE 4 NON-RESIDENT
13981  047416  006662  100000                  MTPI    100000(R2)      ;LOAD TEST DATA INTO PHYSICAL 140000
13982  047422  112767  000006  122660          MOVB    #006,KIPDR4     ;MAKE KERNEL PAGE 4 RESIDENT
```

```
13983  047430  013701  100000          MOV     @#100000,R1     ;READ FROM ADDRESS 140000
13984  047434  020001                  CMP     R0,R1           ;SEE IF DATA WAS STORED CORRECTLY
13985  047436  001401                  BEQ     17$
  (2)  047440  104000                  EMT                     ;INCORRECT STORE
13986                                                          ;FOR TIGHTER SCOPE LOOP
13987                                                          ;REPLACE ERROR CALL WITH
13988                                                          ;"BR 16$" = 000763
13989  047442                  17$:    ;THE FOLLOWING WILL TEST DSTM=7 MTPI.
13990                                  ;
13991  047442  012767  030340  130326  MOV     #030340,PSW     ;MAKE PREVIOUS MODE USER
13992  047450  012700  125252          MOV     #125252,R0      ;LOAD TEST DATA INTO R0
13993  047454  012767  100000  170114  MOV     #100000,$TMP2   ;LOAD VIRT. ADDR. OF TEST LOCATION
13994                                                          ;INTO LOCATION $TMP2
13995  047462  012702  037576          MOV     #$TMP2,R2       ;LOAD ADDRESS OF $TMP2 INTO R2
13996  047466  010046          18$:    MOV     R0,-(KSP)       ;PUSH TEST DATA ON KERNEL STACK
13997  047470  105067  122614          CLRB    KIPDR4          ;MAKE KERNEL PAGE 4 NON-RESIDENT
13998  047474  006672  000000          MTPI    @0(R2)          ;LOAD TEST DATA INTO PHYSICAL 140000
13999  047500  112767  000006  122602  MOVB    #006,KIPDR4     ;MAKE KERNEL PAGE 4 RESIDENT
14000  047506  013701  100000          MOV     @#100000,R1     ;READ FROM ADDRESS 140000
14001  047512  020001                  CMP     R0,R1           ;SEE IF DATA WAS STORED CORRECTLY
14002  047514  001401                  BEQ     19$
  (1)  047516                  20$:
  (2)  047516  104000                  EMT                     ;INCORRECT STORE
14003                                                          ;FOR TIGHTER SCOPE LOOP
14004                                                          ;REPLACE ERROR CALL WITH
14005                                                          ;"BR 18$" = 000763
14006  047520  012767  021550  130522  19$:    MOV     #T0250,MMVEC    ;RESTORE M.M. VECTOR TO NORMAL ROUTINE
14007
14008
14009
14021                                  ;*************************************************************************************
  (2)                                  ;TEST 415       MOVE FROM PREVIOUS (KERNEL) I-SPACE TO USER MODE
  (3)                                  ;*************************************************************************************
  (2)  047526                  TS415:
14022  047526  012700  077406          1$:     MOV     #77406,R0       ;MAKE ALL USER I-SPACE PAGES RESIDENT
14023                                                          ;READ/WRITE, LENGTH 200 BLOCKS
14024  047532  012702  000010          MOV     #10,R2          ;SET LOOP COUNTER TO 8
14025  047536  012701  177600          MOV     #UIPDR0,R1      ;LOAD ADDRESS OF FIRST PDR IN R1
14026  047542  010021          2$:     MOV     R0,(R1)+        ;LOAD PDR WITH 77406
14027  047544  077202                  SOB     R2,2$           ;LOOP UNTIL 8 USER PDRS LOADED
14028  047546  012767  140340  130222  3$:     MOV     #140340,PSW     ;GO TO USER MODE FOR THIS TEST
14029  047554  012767  077406  122526  MOV     #77406,KIPDR4   ;KERNEL I-SPACE PAGE 4 READ/WRITE
14030  047562  012767  001400  122560  MOV     #1400,KIPAR4    ;MAP KERNEL I PAGE 4 TO 24K
14031  047570  012767  001400  130052  MOV     #1400,UIPAR4    ;MAP USER I PAGE 4 TO 24K
14032  047576  012700  036514          MOV     #36514,R0       ;LOAD DATA PATTERN INTO R0
14033  047602  010037  100000          MOV     R0,@#100000     ;LOAD DATA PATTERN INTO PHY 140000
14034  047606  012702  100000          MOV     #100000,R2      ;LOAD VIRTUAL ADDRESS INTO R2
14035                                  ;THE FOLLOWING WILL TEST DSTM=0 MFPI
14036                                  ;
14037  047612  012767  050110  130430  MOV     #21$,MMVEC      ;SET M.M. VECTOR TO 21$
14038  047620  105067  127764          CLRB    UIPDR4          ;MAKE USER I-SPACE PAGE 4 NON-RESIDENT
14039  047624  012767  140340  130144  MOV     #140340,PSW     ;MAKE PREVIOUS MODE KERNEL PRESENT USER
14040  047632  006506          4$:     MFPI    KSP             ;PUT KERNEL STACK POINTER ON USER STACK
14041  047634  022706  000600          CMP     #USESTK,USP     ;WAS SOMETHING PUSHED ON STACK AT 1$
14042  047640  001405                  BEQ     5$              ;BRANCH IF NOTHING WAS PUSHED
14043  047642  012600                  MOV     (USP)+,R0       ;POP USER STACK INTO R0
```

```
14044  047644  012701  001000            MOV     #KERSTK,R1      ;EXPECTING 1100 AS KSP
14045  047650  020001                    CMP     R0,R1           ;DID YOU GET THE RIGHT POINTER?
14046  047652  001401                    BEQ     6$
  (1)  047654                    5$:
  (2)  047654  104000                    EMT                     ;WRONG THING WAS PUSHED ON STACK
14047                                                            ;FOR TIGHTER SCOPE LOOP
14048                                                            ;REPLACE ERROR CALL WITH
14049                                                            ;"BR 4$" = 000766
14050  047656                    6$:     ;THE FOLLOWING WILL TEST DSTM=1 MFPI.
14051  047656  012767  140340  130112  7$:     MOV     #140340,PSW     ;MAKE PREVIOUS MODE KERNEL PRESENT USER
14052  047664  012700  036514            MOV     #36514,R0       ;LOAD DATA EXPECTED INTO R0
14053  047670  012702  100000            MOV     #100000,R2      ;LOAD VIRTUAL ADDRESS INTO R2
14054  047674  006512                    MFPI    (R2)            ;READ FROM PHYSICAL 140000
14055  047676  012601                    MOV     (USP)+,R1       ;POP USER STACK INTO R1
14056  047700  020001                    CMP     R0,R1           ;WAS DATA FETCHED SAME AS STORED
14057  047702  001401                    BEQ     9$
  (2)  047704  104000                    EMT                     ;WRONG DATA WAS FETCHED
14058                                                            ;FOR TIGHTER SCOPE LOOP
14059                                                            ;REPLACE ERROR CALL WITH
14060                                                            ;"BR 7$" = 000764
14061                                   ;THE FOLLOWING WILL TEST DSM=2 MFPI.
14062  047706  012767  140340  130062  9$:     MOV     #140340,PSW     ;MAKE PREVIOUS MODE KERNEL PRESENT USER
14063  047714  012702  100000            MOV     #100000,R2      ;LOAD VIRTUAL ADDRESS INTO R2
14064  047720  006522                    MFPI    (R2)+           ;READ FROM PHYSICAL 140000
14065  047722  012601                    MOV     (USP)+,R1       ;POP USER STACK INTO R1
14066  047724  020001                    CMP     R0,R1           ;WAS DATA FETCHED SAME AS STORED
14067  047726  001401                    BEQ     11$
  (2)  047730  104000                    EMT                     ;WRONG DATA WAS FETCHED
14068                                                            ;FOR TIGHTER SCOPE LOOP
14069                                                            ;REPLACE ERROR CALL WITH
14070                                                            ;"BR 9$" = 000766
14071                                   ;THE FOLLOWING WILL TEST DSTM=3 MFPI.
14072  047732  012767  140340  130036  11$:     MOV     #140340,PSW     ;MAKE PREVIOUS MODE KERNEL PRESENT USER
14073  047740  006537  100000            MFPI    @#100000        ;READ FROM PHYSICAL 140000
14074  047744  012601                    MOV     (USP)+,R1       ;POP USER STACK INTO R1
14075  047746  020001                    CMP     R0,R1           ;WAS DATA FETCHED SAME AS STORED
14076  047750  001401                    BEQ     13$
  (2)  047752  104000                    EMT                     ;WRONG DATA WAS FETCHED
14077                                                            ;FOR TIGHTER SCOPE LOOP
14078                                                            ;REPLACE ERROR CALL WITH
14079                                                            ;"BR 11$" = 000767
14080                                   ;THE FOLLOWING WILL TEST DSTM=4 MFPI.
14081  047754  012767  140340  130014  13$:     MOV     #140340,PSW     ;MAKE PREVIOUS MODE DERNEL PRESENT USER
14082  047762  012702  100002            MOV     #100002,R2      ;LOAD VIRTUAL ADDRESS INTO R2
14083  047766  006542                    MFPI    -(R2)           ;READ FROM PHYSICAL 140000
14084  047770  012601                    MOV     (USP)+,R1       ;POP USER STACK INTO R1
14085  047772  020001                    CMP     R0,R1           ;WAS DATA FETCHED SAME AS STORED
14086  047774  001401                    BEQ     15$
  (2)  047776  104000                    EMT                     ;WRONG DATA WAS FETCHED
14087                                                            ;FOR TIGHTER SCOPE LOOP
14088                                                            ;REPLACE ERROR CALL WITH
14089                                                            ;"BR 13$" = 000766
14090                                   ;THE FOLLOWING WILL TEST DSTM=5 MFPI.
14091                                   ;
14092  050000  012767  140340  127770  15$:     MOV     #140340,PSW     ;MAKE PREVIOUS MODE KERNEL PRESENT USER
14093  050006  012767  100000  167562            MOV     #100000,$TMP2   ;LOAD TEST LOC. VIRT. ADDR INTO LOC. $TMP2
```

```
14094  050014  012702  037600                    MOV     #<$TMP2+2>,R2   ;LOAD ADDRESS OF $TMP2+2 INTO R2
14095  050020  006552                            MFPI    @-(R2)          ;READ FROM PHYSICAL 140000
14096  050022  012601                            MOV     (USP)+,R1       ;POP USER STACK INTO R1
14097  050024  020001                            CMP     R0,R1           ;WAS DATA FETCHED SAME AS STORED
14098  050026  001401                            BEQ     17$
  (2)  050030  104000                            EMT                     ;WRONG DATA WAS FETCHED
14099                                                                    ;FOR TIGHTER SCOPE LOOP
14100                                                                    ;REPLACE ERROR CALL WITH
14101                                                                    ;"BR 15$" = 000763
14102                                          ;THE FOLLOWING WILL TEST DSTM=6 MFPI.
14103                                          ;
14104  050032  012767  140340  127736  17$:     MOV     #140340,PSW     ;MAKE PREVIOUS MODE KERNEL PRESENT USER
14105  050040  005002                            CLR     R2              ;MAKE REGISTER 2 A ZERO
14106  050042  006562  100000                    MFPI    100000(R2)      ;READ FROM PHYSICAL 140000
14107  050046  012601                            MOV     (USP)+,R1       ;POP USER STACK INTO R1
14108  050050  020001                            CMP     R0,R1           ;WAS DATA FETCHED SAME AS STORED
14109  050052  001401                            BEQ     19$
  (2)  050054  104000                            EMT                     ;WRONG DATA WAS FETCHED
14110                                                                    ;FOR TIGHTER SCOPE LOOP
14111                                                                    ;REPLACE ERROR CALL WITH
14112                                                                    ;"BR 17$" = 000766
14113                                          ;THE FOLLOWING WILL TEST DSTM=7 MFPI.
14114                                          ;
14115  050056  012767  140340  127712  19$:     MOV     #140340,PSW     ;MAKE PREVIOUS MODE KERNEL PRESENT USER
14116  050064  012767  100000  167504            MOV     #100000,$TMP2   ;LOAD TEST LOC. VIRT. ADDR. INTO $TMP2
14117  050072  012702  037576                    MOV     #$TMP2,R2       ;LOAD ADDRESS OF $TMP2 INTO R2
14118  050076  006572  000000                    MFPI    @0(R2)          ;READ FROM PHYSICAL 140000
14119  050102  012601                            MOV     (USP)+,R1       ;POP USER STACK INTO R1
14120  050104  020001                            CMP     R0,R1           ;WAS DATA FETCHED SAME AS STORED
14121  050106  001401                            BEQ     20$
  (1)  050110                            21$:
  (2)  050110  104000                            EMT                     ;WRONG DATA WAS FETCHED
14122                                                                    ;FOR TIGHTER SCOPE LOOP
14123                                                                    ;REPLACE ERROR CALL WITH
14124                                                                    ;"BR 19$" = 000762
14125  050112  012767  021550  130130  20$:     MOV     #T0250,MMVEC    ;SET M.M. VECTOR TO NORMAL ROUTINE
14126  050120  012767  000340  127650            MOV     #00340,PSW      ;GO BACK TO KERNEL MODE, PREVIOUS KERNEL
14127
14135                                          ;*****************************************************************************
  (2)                                          ;TEST 416       MOVE FROM/TO D-SPACE = MOVE FROM/TO I-SPACE
  (3)                                          ;*****************************************************************************
  (2)  050126                                  TS416:
14136  050126  012767  030340  127642  1$:      MOV     #030340,PSW     ;MAKE PREVIOUS MODE=USER,CURRENT=KERNEL
14137  050134  106506                            MFPD    USP             ;MFPD SHOULD ACT LIKE MFPI PUTTING
14138                                                                    ;USER STACK POINTER ON THE KERNEL STACK
14139  050136  022706  001000                    CMP     #KERSTK,KSP     ;WAS SOMETHING PUSHED ON KERNEL STACK?
14140  050142  001405                            BEQ     2$              ;BRANCH IF NO
14141  050144  012600                            MOV     (KSP)+,R0       ;POP KERNEL STACK INTO R0
14142  050146  012701  000600                    MOV     #USESTK,R1      ;EXPECTING TO GET 700 AS USP
14143  050152  020001                            CMP     R0,R1           ;DID GET RIGHT POINTER VALUE?
14144  050154  001401                            BEQ     4$
  (1)  050156                            2$:
  (2)  050156  104000                            EMT                     ;WRONG THING WAS PUSHED ON STACK
14145                                                                    ;FOR TIGHTER SCOPE LOOP
14146                                                                    ;REPLACE ERROR CALL WITH
14147                                                                    ;"BR 1$" = 000763
```

```
14148  050160  012746  007777          4$:     MOV     #7777,-(KSP)    ;PUSH DATA ON KERNEL STACK
14149  050164  106606                          MTPD    USP             ;LOAD THE USER STACK POINTER
14150  050166  106506                          MFPD    USP             ;READ USER STACK POINTER
14151  050170  012601                          MOV     (KSP)+,R1       ;POP KERNEL STACK INTO R1
14152  050172  022701  007777                  CMP     #7777,R1        ;WAS USER STACK POINTER CHANGED?
14153  050176  001401                          BEQ     5$
  (2)  050200  104000                          EMT                     ;USER STACK POINTER NOT CHANGED
14154                                                                   ;FOR TIGHTER SCOPE LOOP
14155                                                                   ;REPLACE ERROR CALL WITH
14156                                                                   ;"BR 4$" = 000767
14157  050202  012746  000600          5$:     MOV     #USESTK,-(KSP)  ;GET READY TO RESTORE USER STK. PTR.
14158  050206  106606                          MTPD    USP             ;RESTORE USER STACK POINTER
14159
14160                                  ;*****************************************************************************
  (2)                                  ;TEST 417       MOVE FROM PREVIOUS I-SPACE (PREVIOUS=CURRENT=KERNEL)
  (3)                                  ;*****************************************************************************
  (2)  050210                          TS417:
14161  050210  005037  177776          1$:     CLR     @#PSW           ;SET PREVIOUS = CURRENT = KERNEL
14162  050214  012700  001000                  MOV     #KERSTK,R0      ;SETUP VALUE FOR STACK POINTER
14163  050220  010006                          MOV     R0,KSP          ;LOAD STACK POINTER
14164  050222  006506                          MFPI    KSP             ;THE VALUE "STACK" SHOULD BE PUSHED
14165                                                                   ;BEFORE BEING DECREMENTED
14166  050224  011601                          MOV     (KSP),R1        ;READ DATA WHICH WAS PUSHED
14167  050226  020001                          CMP     R0,R1           ;WAS THE ORIGINAL VALUE OF THE
14168                                                                   ;STACK POINTER PUSHED?
14169  050230  001401                          BEQ     2$
  (2)  050232  104000                          EMT                     ;MFPI FETCHED WRONG DATA
14170                                                                   ;FOR TIGHTER SCOPE LOOP
14171                                                                   ;REPLACE ERROR CALL WITH
14172                                                                   ;"BR 1$" = 000766
14173  050234  005740                  2$:     TST     -(R0)           ;SETUP EXPECTED STACK POINTER VALUE
14174  050236  020600                          CMP     KSP,R0          ;WAS THE STACK POINTER DECREMENTED?
14175  050240  001401                          BEQ     3$
  (2)  050242  104000                          EMT                     ;STACK NOT PUSHED BY THE MFPI
14176                                                                   ;FOR TIGHTER SCOPE LOOP
14177                                                                   ;REPLACE ERROR CALL WITH
14178                                                                   ;"BR 1$" = 000762
14179  050244  012706  001000          3$:     MOV     #KERSTK,KSP     ;RESTORE STACK POINTER
14180  050250  005067  127316                  CLR     SR0             ;TURN OFF MEMORY MANAGEMENT UNIT
14181  050254  000167  000476                  JMP     EXATST          ;GET OVER SUBROUTINES TO EXTENDED ADRS TESTS
```

```
                                    .SBTTL  *****  SUBROUTINES USED BY THIS PROGRAM  *****

                                    .SBTTL  TURN OFF T-BIT AND SAVE CURRENT PSW
                                    ;*****************************************************************************
                                    ;*
                                    ;*      THIS SUBROUTINE IS USED TO TURN OFF THE TRACE TRAP BIT IN THE PSW
                                    ;*      IF IT IS ON.  THE PROCESSOR STATUS IS SAVED IN "TBITPS" SO THAT
                                    ;*      THE PSW CAN BE RESTORED TO ITS PREVIOUS CONDITION WHEN CONDITIONS
                                    ;*      WARRANT T-BIT TRAPPING.
                                    ;*
                                    ;*****************************************************************************
050260  036727  127512  000020  TOFF:  BIT     PSW,#TBIT       ;IS THE T-BIT SET IN THE PSW?
050266  001411                       BEQ     1$              ;EXIT IF NO
050270  016746  127502               MOV     PSW,-(SP)       ;PUSH PRESENT PSW ON THE STACK
050274  011667  167270               MOV     (SP),TBITPS     ;ALSO SAVE IT IN "TBITPS" FOR
                                                             ;RESTORING LATER
050300  042716  000020               BIC     #TBIT,(SP)      ;CLEAR THE T-BIT (BIT 4) IN THE PSW
050304  012746  050312               MOV     #1$,-(SP)       ;PUSH PC OF "RTS" ON STACK
050310  000006                       RTT                     ;"RETURN" TO 1$ WITH T-BIT OFF
050312  000207                 1$:   RTS     PC              ;RETURN TO PROGRAM

                                    .SBTTL  TURN ON T-BIT AND RESTORE PREVIOUS PSW
                                    ;*****************************************************************************
                                    ;*
                                    ;*      THIS SUBROUTINE IS USED TO RESTORE THE PROCESSOR STATUS TO ITS
                                    ;*      PREVIOUS CONDITION BY RESTORING THE "T-BIT PSW" SAVED BY THE
                                    ;*      "TOFF" SUBROUTINE IN THE "TBITPS" LOCATION.
                                    ;*
                                    ;*****************************************************************************
050314  036727  167250  000020  TON:   BIT     TBITPS,#TBIT    ;WAS T-BIT ON IN THE PREVIOUS PSW?
050322  001410                       BEQ     1$              ;EXIT IF NO
050324  016746  167240               MOV     TBITPS,-(SP)    ;PUSH PREVIOUS PSW ON THE STACK
050330  012767  000340  167232       MOV     #340,TBITPS     ;RESET THE "TBITPS" LOCATION
050336  012746  050344               MOV     #1$,-(SP)       ;PUSH PC OF "RTS" ON STACK
050342  000006                       RTT                     ;"RETURN" TO 1$ WITH T-BIT RESTORED
050344  000207                 1$:   RTS     PC              ;RETURN TO PROGRAM

                                    .SBTTL  SET ALL WRITEABLE BITS IN ALL PAR/PDR'S
                                    ;*****************************************************************************
                                    ;*
                                    ;*      THIS SUBROUTINE IS USED BY THE PAR/PDR DUAL ADDRESSING TEST
                                    ;*      TO SET ALL WRITEABLE BITS IN ALL KERNEL AND USE PAR'S AND
                                    ;*      PDR'S TO A 1.  THE "INITIAL STATE" OF HAVING ALL BITS=1 IS
                                    ;*      USED TO SEE THAT ONLY ONE REGISTER IS CLEARED IN RESPONSE TO
                                    ;*      A SINGLE PAR OR PDR ADDRESS.
                                    ;*
                                    ;*****************************************************************************
050346  012702  000010  SETREG: MOV     #10,R2          ;LOAD LOOP COUNTER WITH AN 8
050352  012701  172300           MOV     #KIPDR0,R1      ;LOAD ADDRESS OF FIRST PDR INTO R1
050356  012721  177777     1$:   MOV     #-1,(R1)+       ;SET BITS IN KERNEL PDR TO 1
050362  077203                   SOB     R2,1$           ;LOOP TO 1$ UNTIL ALL KERNEL PDR'S LOADED
050364  012702  000010           MOV     #10,R2          ;LOAD LOOP COUNTER WITH AN 8
050370  012701  172340           MOV     #KIPAR0,R1      ;LOAD ADDRESS OF FIRST PAR INTO R1
050374  012721  177777     2$:   MOV     #-1,(R1)+       ;SET BITS IN A KERNEL PAR TO 1
050400  077203                   SOB     R2,2$           ;LOOP TO 2$ UNTIL ALL KERNEL PAR'S LOADED
```

```
14247  050402  012702  000010                MOV     #10,R2          ;LOAD LOOP COUNTER WITH AN 8
14248  050406  012701  177600                MOV     #UIPDR0,R1      ;LOAD ADDRESS OF FIRST PDR INTO R1
14249  050412  012721  177777        3$:     MOV     #-1,(R1)+       ;SET BITS IN A USER PDR TO 1
14250  050416  077203                        SOB     R2,3$           ;LOOP TO 3$ UNTIL ALL USER PDR'S LOADED
14251  050420  012702  000010                MOV     #10,R2          ;LOAD LOOP COUNTER WITH AN 8
14252  050424  012701  177640                MOV     #UIPAR0,R1      ;LOAD ADDRESS OF FIRST PAR INTO R1
14253  050430  012721  177777        4$:     MOV     #-1,(R1)+       ;SET BITS IN A USER PAR TO 1
14254  050434  077203                        SOB     R2,4$           ;LOOP TO 4$ UNTIL ALL USER PAR'S LOADED
14255  050436  000207                        RTS     PC              ;RETURN TO TEST
14256
14257                                  .SBTTL  READ & COMPARE KERNEL & USER PAR/PDR'S
14258                                  ;*****************************************************************************
14259                                  ;*
14260                                  ;*      THIS SUBROUTINE IS USED BY PAR/PDR DUAL ADDRESSING TEST TO
14261                                  ;*      READ ALL THE PAR'S AND PDR'S TO SEE THAT ONLY ONE REGISTER
14262                                  ;*      WAS CLEARED IN RESPONSE TO A SINGLE PAR OR PDR ADDRESS.
14263                                  ;*      ANY FAILURES FOUND BY THE PAR/PDR DUAL ADDRESSING TEST WILL
14264                                  ;*      BE REPORTED BY THIS SUBROUTINE.
14265                                  ;*
14266                                  ;*****************************************************************************
14267  050440                          CMPREG:
14268  050440  012701  172300                MOV     #KIPDR0,R1      ;LOAD ADDRESS OF FIRST KERNEL PDR IN R1
14269  050444  012704  000010                MOV     #10,R4          ;LOAD LOOP COUNTER WITH AN 8
14270  050450  012705  077416                MOV     #77416,R5       ;PUT EXPECTED PDR CONTENTS IN R5
14271  050454  021105                1$:     CMP     (R1),R5         ;ARE ALL WRITEABLE BITS SET AS EXPECTED?
14272  050456  001403                        BEQ     2$              ;BRANCH IF YES
14273  050460  020100                        CMP     R1,R0           ;WAS IT THE REG. THAT WAS CLEARED?
14274  050462  001401                        BEQ     2$
  (2)  050464  104000                        EMT                     ;A PDR WAS EFFECTED BY CLEARING A DIFFERENT PAR/PRD
14275                                                                  ;FOR TIGHTER SCOPE LOOP
14276                                                                  ;REPLACE ERROR CALL WITH
14277                                                                  ;AN "RTS PC" = 000207
14278  050466  062701  000002        2$:     ADD     #2,R1           ;FORM NEXT ADDRESS
14279  050472  077410                        SOB     R4,1$           ;LOOP TO 1$ UNTIL ALL KERNEL PDR'S CHECKED
14280  050474  012701  172340                MOV     #KIPAR0,R1      ;LOAD ADDRESS OF FIRST KERNEL PAR IN R1
14281  050500  012704  000010                MOV     #10,R4          ;LOAD LOOP COUNTER WITH AN 8
14282                                                                  ;****F11 CHANGE**** FROM #7777 TO #177777
14283  050504  012705  177777                MOV     #177777,R5      ;PUT EXPECTED PAR CONTENTS IN R5
14284  050510  021105                3$:     CMP     (R1),R5         ;ARE ALL WRITEABLE BITS SET AS EXPECTED?
14285  050512  001403                        BEQ     4$              ;BRANCH IF YES
14286  050514  020100                        CMP     R1,R0           ;WAS IT THE REG. THAT WAS CLEARED?
14287  050516  001401                        BEQ     4$
  (2)  050520  104000                        EMT                     ;A PAR WAS EFFECTED BY CLEARING A DIFFENENT PAR/PDR
14288                                                                  ;FOR TIGHTER SCOPE LOOP
14289                                                                  ;REPLACE ERROR CALL WITH
14290                                                                  ;AN "RTS PC" = 000207
14291  050522  062701  000002        4$:     ADD     #2,R1           ;FORM NEXT ADDRESS
14292  050526  077410                        SOB     R4,3$           ;LOOP TO 3$ UNTIL ALL KERNEL PAR'S CHECKED
14293  050530  012701  177600                MOV     #UIPDR0,R1      ;LOAD ADDRESS OF FIRST USER PDR IN R1
14294  050534  012704  000010                MOV     #10,R4          ;LOAD LOOP COUNTER WITH AN 8
14295  050540  012705  077416                MOV     #77416,R5       ;PUT EXPECTED PDR CONTENTS IN R5
14296  050544  021105                5$:     CMP     (R1),R5         ;ARE ALL WRITEABLE BITS SET AS EXPECTED?
14297  050546  001403                        BEQ     6$              ;BRANCH IF YES
14298  050550  020100                        CMP     R1,R0           ;WAS IT THE REG. THAT WAS CLEARED?
14299  050552  001401                        BEQ     6$
  (2)  050554  104000                        EMT                     ;A PDR WAS EFFECTED BY CLEARING A DIFFERENT PAR/PDR
```

```
14300                                                                   ;FOR TIGHTER SCOPE LOOP
14301                                                                   ;REPLACE ERROR CALL WITH
14302                                                                   ;AN "RTS PC" = 000207
14303  050556  062701  000002          6$:     ADD     #2,R1          ;FORM NEXT ADDRESS
14304  050562  077410                          SOB     R4,5$          ;LOOP TO 5$ UNTIL ALL USER PDR'S CHECKED
14305  050564  012701  177640                  MOV     #UIPAR0,R1     ;LOAD ADDRESS OF FIRST USER PAR IN R1
14306  050570  012704  000010                  MOV     #10,R4         ;LOAD LOOP COUNTER WITH AN 8
14307                                                                  ;****F11 CHANGE**** FROM #7777 TO #177777
14308  050574  012705  177777                  MOV     #177777,R5     ;PUT EXPECTED PAR CONTENTS IN R5
14309  050600  021105                  7$:     CMP     (R1),R5        ;ARE ALL WRITEABLE BITS SET AS EXPECTED?
14310  050602  001403                          BEQ     8$             ;BRANCH IF YES
14311  050604  020100                          CMP     R1,R0          ;WAS IT THE REG. THAT WAS CLEARED?
14312  050606  001401                          BEQ     8$
  (2)  050610  104000                          EMT                    ;A PAR WAS EFFECTED BY CLEARING A DIFFERENT PAR/PDR
14313                                                                   ;FOR TIGHTER SCOPE LOOP
14314                                                                   ;REPLACE ERROR CALL WITH
14315                                                                   ;AN "RTS PC" = 000207
14316  050612  062701  000002          8$:     ADD     #2,R1          ;FORM NEXT ADDRESS
14317  050616  077410                          SOB     R4,7$          ;LOOP TO 7$ UNTIL ALL USER PAR'S CHECKED
14318  050620  000207                          RTS     PC             ;RETURN TO TEST
14319
14320                                  .SBTTL  INHIBIT 'RESETS' WHILE UNDER APT
14321                                  ;*****************************************************************************
14322                                  ;*
14323                                  ;*      THIS SUBROUTINE CONTROLS THE USAGE OF RESET INST'S WHILE
14324                                  ;*      RUNNING UNDER APT. RESETS ARE ALLOWED DURING THE FIRST
14325                                  ;*      PASS OF THE DIAGNOSTIC.
14326                                  ;*
14327  050622  126727  130172  000001  CHKAPT: CMPB    $ENV,#1        ;ARE WE RUNNING UNDER APT?
14328  050630  001003                          BNE     1$             ;NO BRANCH
14329  050632  005767  130150                  TST     $PASS          ;IS THIS THE FIRST PASS?
14330  050636  001002                          BNE     RETA           ;NO BRANCH
14331  050640  062705  000002          1$:     ADD     #2,R5          ;BUMP RETURN ADDRESS FOR NORMAL TESTING
14332  050644  000205                  RETA:   RTS     R5             ;RETURN
14333
14334
14335                                  .SBTTL  ERROR ROUTINE FOR MEMORY MANAGEMENT TEST
14336                                  ;*****************************************************************************
14337                                  ;*      THIS IS THE ONLY ERROR REPORT FOR ALL THE MMU TESTS
14338                                  ;*****************************************************************************
14339
14340  050646  004767  062556          ERROR3: JSR     PC,ABORT               ;ARE WE UNDER UFD ?
14341  050652  012737  000002  001002          MOV     #2,@#$FATAL            ;SET UP FATAL ERROR NUMBER
14342  050660  012767  000001  130112          MOV     #1,$MSGTY              ;SET FATAL ERROR FLAG
14343  050666  032737  000001  001020          BIT     #1,@#$ENV              ;UNDER APT ?
14344  050674  001004                          BNE     MMUHLT
14345  050676  012700  050710                  MOV     #MMUMSG,R0
14346  050702  004767  062456                  JSR     PC,TYPE
14347  050706  000777                  MMUHLT: BR      .                      ;STAY IN LOOP
14348
14349  050710  040506  046111  042105  MMUMSG: .ASCIZ  /FAILED DURING MMU TESTING/<12><15>
       050716  042040  051125  047111
       050724  020107  046515  020125
       050732  042524  052123  047111
       050740  005107  000015
14350
```

F2



```
14351                                   .EVEN
14352
14353
14354                                   ;******************** **************************************************
14355                                   ;
14356                                   ;       STORAGE AREAS FOR THE FOLLOWING FEW TESTS
14357                                   ;
14358                                   ;*****************************************************************************
14359
14360   050744  000000                  TMP:    .WORD   0
14361   050746  000000                  TMP1:   .WORD   0
14362   050750  000000                  MEM:    .WORD   0
14363   050752  000002                  CNTR:   .BLKW   2
14364                                   ;*************************************************************************************************
  (2)                                   ;TEST 420       TEST ADDRESS BITS 17-21
  (3)                                   ;*************************************************************************************************
  (2)   050756                          TS420:
14365                                   ;*****************************************************************
14366                                   ;
14367                                   ;       THIS TEST WILL DETERMINE WHETHER THE KDF11-B CAN
14368                                   ;       DRIVE ADDRESS BITS 17-21.
14369                                   ;
14370                                   ;******* ********************************************************
14371
14372   050756  012767  177600  121372  EXATST: MOV     #177600,KIPAR7          ;I O PAGE
14373   050764  012737  000007  001004          MOV     #7,@#$TESTN
14374   050772  012737  051550  000030          MOV     #ERRORA,@#30
14375   051000  000244                          CLZ                             ;CLR THE Z BIT
14376   051002  032777  000200  150566          BIT     #200,@SWR               ;WANT TO TEST THIS ?
14377   051010  001001                          BNE     ESR3
14378   051012  000470                          BR      NEXT
14379
14380   051014  016767  126764  177722  ESR3:   MOV     ERRVEC,TMP
14381   051022  016767  126760  177716          MOV     ERRVEC+2,TMP1
14382   051030  012767  051132  126746          MOV     #ESR0,ERRVEC            ;PREPARE FOR NEW
14383   051036  012767  000340  126742          MOV     #340,ERRVEC+2           ;INTERRUPT
14384   051044  012767  004000  121276          MOV     #4000,KIPAR4            ;SET FOR ADRS BIT 17
14385   051052  012767  000001  126512          MOV     #BIT0,SR0
14386   051060  012767  000020  121430          MOV     #BIT4,SR3               ;TURN 22 BIT ADRSG AND
14387                                                                           ;MEMORY MANG ON
14388   051066  012767  000000  177656          MOV     #0,CNTR
14389   051074  026727  177652  000005  ESR1:   CMP     CNTR,#5
14390   051102  001420                          BEQ     ESR2
14391   051104  013737  100000  100000          MOV     @#100000,@#100000       ;TRY ADRSG IT
14392   051112  000240                          NOP
14393   051114  012767  000000  126450          MOV     #0,SR0
14394   051122  012767  000000  121366          MOV     #0,SR3                  ;TURN OFF MM AND 22 BIT
14395   051130  104000                          EMT                             ;ERROR
14396   051132  006367  121212          ESR0:   ASL     KIPAR4                  ;SHOULD TIME OUT
14397   051136  005267  177610                  INC     CNTR                    ;SHIFT 1 TO THE LEFT
14398   051142  000754                          BR      ESR1
14399   051144  016767  177574  126632  ESR2:   MOV     TMP,ERRVEC
14400   051152  016767  177570  126626          MOV     TMP1,ERRVEC+2           ;RESTORE
14401   051160  012767  000000  126404          MOV     #0,SR0
14402   051166  012767  000000  121322          MOV     #0,SR3                  ;TURN OFF MM AND 22 BIT
14403
```

```
14404
14405                                   ;*****************************************************************************
  (2)                                   ;TEST 421       TEST ADDRESS BIT 16
  (3)                                   ;*****************************************************************************
  (2)  051174                           TS421:
14406                                   ;*************************************************************
14407                                   ;
14408                                   ;       THIS TEST WILL CHECK TO SEE IF THE 17TH BIT (16) OF AN 18 BIT
14409                                   ;       ADDRESS CAN BE DRIVEN.
14410                                   ;
14411                                   ;*************************************************************
14412
14413  051174  012767  000001  126370  NEXT:   MOV     #BIT0,SR0                       ;TURN ON MEM MANG
14414  051202  016767  126576  177534          MOV     ERR.EC,TMP
14415  051210  016767  126572  177530          MOV     ERRVEC+2,TMP1
14416  051216  012767  051300  126560          MOV     #ESR4,ERRVEC
14417  051224  012767  000340  126554          MOV     #340,ERRVEC+2
14418  051232  012737  123456  000000          MOV     #123456,@#0                     ;MODIFY LOCATION ZERO
14419  051240  012767  002000  121102          MOV     #2000,KIPAR4
14420  051246  013737  100000  100000          MOV     @#100000,@#100000
14421  051254  023727  000000  123456          CMP     @#0,#123456                     ;CHANGED ?
14422  051262  001407                          BEQ     ESR5
14423  051264  012767  000000  126300          MOV     #0,SR0
14424  051272  012767  000000  121216          MOV     #0,SR3                  ;TURN OFF MM AND 22 BIT
14425  051300  104000                  ESR4:   EMT                             ;LOC 0 WAS ALTERED
14426  051302  005067  126264          ESR5:   CLR     SR0                     ;TURN OFF MM
14427  051306  016767  177432  126470          MOV     TMP,ERRVEC
14428  051314  016767  177426  126464          MOV     TMP1,ERRVEC+2                   ;RESTORE VECTORS
14429
14430
14431
14432                                   ;*****************************************************************************
  (2)                                   ;TEST 422       TEST PARITY ERROR DETECTION LOGIC
  (3)                                   ;*****************************************************************************
  (2)  051322                           TS422:
14433
14434                                   ;*************************************************************
14435                                   ;
14436                                   ;       THIS TEST WILL USE THE MEMORY PARITY CSR TO GENERATE
14437                                   ;       A PARITY ERROR THAT THE CPU SHOULD DETECT.
14438                                   ;
14439                                   ;*************************************************************
14440
14441
14442  051322  000244                          CLZ                             ;CLR THE Z BIT
14443  051324  032777  001000  150244          BIT     #BIT9,@SWR              ;TEST THIS ?
14444  051332  001002                          BNE     ESR57
14445  051334  000167  000434                  JMP     @22TST                  ;NO
14446
14447  051340  012767  000001  000374  ESR57:  MOV     #1,ERRORE               ;DUMMY BIT
14448  051346  016767  126432  177376          MOV     ERRVEC,CNTR
14449  051354  016767  126426  177372          MOV     ERRVEC+2,CNTR+2         ;SAVE VECTORS
14450  051362  012767  051400  126414          MOV     #2$,ERRVEC              ;NEW TIME OUT VECOTR
14451  051370  012704  172100                  MOV     #172100,R4              ;PLACE TO START
14452  051374  005714                  1$:     TST     (R4)                    ;IS IT THERE
14453  051376  000407                          BR      ESR7
```

H2



```
14454  051400  022704  172136          2$:     CMP     #172136,R4              ;TOP YET ?
14455  051404  100403                          BMI     ESR6
14456  051406  062704  000002                  ADD     #2,R4
14457  051412  000770                          BR      1$
14458  051414  104000                  ESR6:   EMT                             ;COULD NOT FIND MEM PAR CSR
14459  051416  016767  126472  177320  ESR7:   MOV     114,TMP                 ;SAVE CONTENTS OF 114
14460  051424  012767  051472  126462          MOV     #ESR8,114               ;PARITY TRAP VECTOR
14461  051432  012714  000005                  MOV     #5,(R4)                 ;BITS 0,2 IN MEM CSR
14462  051436  010167  177306                  MOV     R1,MEM
14463  051442  042714  000004                  BIC     #4,(R4)                 ;TURN OFF WRT WRG PAR BIT
14464  051446  016701  177276                  MOV     MEM,R1                  ;SOME DATA  TO MEMORY FROM
14465  051452  012714  000000          ESR9:   MOV     #0,(R4)                 ;TURN OFF P ERR
14466  051456  010167  177266                  MOV     R1,MEM
14467  051462  016767  177256  126424          MOV     TMP,114                 ;RESTORE VECTOR CONTENTS
14468                                                                          ;CPU, SHOULD SET A PAR ERR
14469  051470  104000                          EMT
14470  051472  032714  100000          ESR8:   BIT     #BIT15,(R4)             ;SEE IF BIT 15 SET IN MEM CSR
14471  051476  001765                          BEQ     ESR9                    ;Z IS SET IF AN ERROR
14472  051500  005000                          CLR     R0
14473  051502  005200                  ESR19:  INC     R0                      ;TIME FOR LED TO BE SEEN
14474  051504  005700                          TST     R0
14475  051506  100375                          BPL     ESR19
14476  051510  012714  000000                  MOV     #0,(R4)
14477  051514  010167  177230                  MOV     R1,MEM
14478  051520  016767  177220  126366          MOV     TMP,114
14479  051526  016767  177220  126250          MOV     CNTR,ERRVEC
14480  051534  016767  177214  126244          MOV     CNTR+2,ERRVEC+2
14481  051542  005067  000174                  CLR     ERRORE                  ;RESTORE DUMMY BIT
14482  051546  000512                          BR      Q22TST                  ;GO FIND A Q22BE
14483                                  ;*************************************************************************
14484
14485  051550  004767  061654          ERRORA: JSR     PC,ABORT                ;ARE WE UNDER UFD ?
14486  051554  012737  000007  001002          MOV     #7,@#$FATAL
14487  051562  012767  000001  127210          MOV     #1,$MSGTY
14488  051570  032737  000001  001020          BIT     #1,@#$ENV               ;UNDER APT ?
14489  051576  001013                          BNE     EXADHT
14490  051600  026727  000136  000001          CMP     ERRORE,#1               ;DUMMY BIT THERE  ?
14491  051606  001003                          BNE     ESR34
14492  051610  012700  051673                  MOV     #CSRMSG,R0
14493  051614  000402                          BR      ESR34+4
14494  051616  012700  051630          ESR34:  MOV     #EXTMSG,R0
14495  051622  004767  061536                  JSR     PC,TYPE
14496  051626  000777                  EXADHT: BR      .
14497
14498
14499  051630  040506  046111  042105  EXTMSG: .ASCIZ  /FAILED DURING EXTENDED ADRS TEST/<12><15>
       051636  042040  051125  047111
       051644  020107  054105  042524
       051652  042116  042105  040440
       051660  051104  020123  042524
       051666  052123  006412     000
14500
14501  051673     106  044501  042514  CSRMSG: .ASCIZ  /FAILED MEM PARITY ERROR DETECT TEST/<12><15>
       051700  020104  042515  020115
       051706  040520  044522  054524
       051714  042440  051122  051117
```

```
        051722  042040  052105  041505
        051730  020124  042524  052123
        051736  006412     000

14502
14503           051742                  .EVEN
14504   051742  000000                  ERRORE: .WORD   0               ;DUMMY BIT
14505
14506                                   ;*****************************************************************
14507   051744  000510                  VECT1:  .WORD   510             ;FIRST DEV VECTOR Q22BE
14508   051746  000000                  DEVECT: .WORD   0
14509   051750  170000                  DEV1:   .WORD   170000                  ;FIRST DEV ADRS Q22BE
14510   051752  000000                  DEVADR: .WORD   0
14511   051754  000000                  CSR1:   .WORD   0
14512   051756  000000                  CSR2:   .WORD   0
14513   051760  000000                  BA:     .WORD   0
14514   051762  000000                  WC:     .WORD   0
14515   051764  000000                  DATA:   .WORD   0
14516   051766  000000                  LATCNT: .WORD   0
14517   051770  000000                  MVLCNT: .WORD   0
14518   051772  000000                  SIMGOA: .WORD   0
14519
14520
14521
14522                                   ;*************************************************************************************
  (2)                                   ;TEST 423       SEE IF A Q22BE(QBE) IS THERE
  (3)                                   ;*************************************************************************************
  (2)   051774                          TS423:
14523
14524                                   ;*************************************************************
14525                                   ;
14526                                   ;       ROUTINE TO SIZE FOR THE Q22BE(QBE) DEVICE ADDRESS
14527                                   ;       WE WILL LOOK FOR A Q22BE(QBE). IF IT ISN'T THERE
14528                                   ;       WE WILL CAUSE AN ERROR VIA THE EMT INSTRUCTION.
14529                                   ;
14530                                   ;*************************************************************
14531
14532   051774  000244                  Q22TST: CLZ
14533   051776  032777  000100  147572          BIT     @#BIT6,@SWR
14534   052004  001002                        BNE     1$
14535   052006  000167  001502                JMP     @DVTST
14536   052012  012737  000011  001004  1$:   MOV     #11,@#TESTN             ;TEST NUM IN MAILBOX
14537   052020  012737  053402  000030        MOV     #ERRORB,@#30            ;SET UP FOR CORRECT EMT
14538
14539
14540   052026  005067  125540                CLR     SR0
14541   052032  016767  125746  176710        MOV     ERRVEC,MEM              ;STORE ERRVEC CONTENTS
14542   052040  012767  052230  125736        MOV     #ESR99,ERRVEC
14543   052046  016767  177676  177676        MOV     DEV1,DEVADR
14544   052054  016767  177664  177664        MOV     VECT1,DEVECT
14545   052062  005777  177664          ESR11: TST     @DEVADR         ;SEE IF IT RESPONDS
14546
14547   052066  016767  177660  177660        MOV     DEVADR,CSR1
14548   052074  016767  177654  177654        MOV     CSR1,CSR2
14549   052102  062767  000002  177646        ADD     #2,CSR2                 ;YES IT DID
14550   052110  016767  177640  177642        MOV     CSR1,BA
14551   052116  062767  000004  177634        ADD     #4,BA
```

J2



```
14552  052124  016767  177624  177630          MOV     CSR1,WC
14553  052132  062767  000006  177622          ADD     #6,WC
14554  052140  016767  177610  177616          MOV     CSR1,DATA
14555  052146  062767  000010  177610          ADD     #10,DATA
14556  052154  016767  177574  177604          MOV     CSR1,LATCNT
14557  052162  062767  000012  177576          ADD     #12,LATCNT
14558  052170  016767  177560  177572          MOV     CSR1,MVLCNT
14559  052176  062767  000014  177564          ADD     #14,MVLCNT
14560  052204  016767  177544  177560          MOV     CSR1,SIMGOA
14561  052212  062767  000016  177552          ADD     #16,SIMGOA
14562  052220  016767  176524  125556          MOV     MEM,ERRVEC              ;RESTORE ERROR VECTOR
14563  052226  000413                          BR      ESR98
14564  052230  062767  000020  177514  ESR99:  ADD     #20,DEVADR
14565  052236  062767  000004  177502          ADD     #4,DEVECT
14566  052244  026727  177476  000550          CMP     DEVECT,#550
14567  052252  001303                          BNE     ESR11                   ;GO TRY ANOTHER ADRS
14568  052254  104000                          EMT
14569  052256  016700  177464  ESR98:  MOV     DEVECT,R0
14570  052262  062700  000002                  ADD     #2,R0
14571  052266  012710  000340                  MOV     #340,(R0)
14572
14573
14574                          ;*************************************************************************
  (2)                          ;TEST 424       USE Q22BE(QBE) TO ALTER THE INTERRUPT LEVEL BITS
  (3)                          ;*************************************************************************
  (2)  052272                  TS424:
14575
14576                          ;*************************************************************
14577                          ;
14578                          ;       GENERATE A Q22BE(QBE) SOFTWARE INTR AT LEVEL 4.
14579                          ;       THE FOLLOWING CODE WILL USE THE Q22(QBE) BUS EXERCISER
14580                          ;       TO GENERATE INTERRUPTS THAT THE CPU SHOULD EITHER
14581                          ;       HONOR OR IGNORE DEPENDING ON THE INTR LEVEL BITS.
14582                          ;
14583                          ;*************************************************************
14584
14585
14586  052272  012777  052336  177446          MOV     #ESR21,@DEVECT          ;INTERRUPT TO ESR21
14587  052300  012777  000001  177446          MOV     #1,@CSR1
14588  052306  106427  000140                  MTPS    #140            ;CPU AT LEVEL THREE
14589  052312  012777  000003  177436          MOV     #3,@CSR2
14590  052320  000240                          NOP
14591  052322  012777  000002  177426          MOV     #2,@CSR2
14592  052330  000240                          NOP
14593  052332  000240                          NOP
14594  052334  104000                          EMT
14595  052336  012777  052372  177402  ESR21:  MOV     #ESR22,@DEVECT          ;CHANGE INTR VECTOR
14596  052344  106427  000200                  MTPS    #200
14597  052350  012777  000003  177400          MOV     #3,@CSR2
14598  052356  000240                          NOP
14599  052360  012777  000000  177370          MOV     #0,@CSR2                ;INTR LEVEL IS 4
14600  052366  000240                          NOP                             ;INTR SHOULD NOT HONORED
14601  052370  000401                          BR      ESR23
14602  052372  104000                  ESR22:  EMT                             ;INTR HONORED A ERROR
14603                          ;*************************************************************
14604                          ;       GENERATE AN INTERRUPT AT LEVEL FIVE
```

```
14605                                   ;*****************************************************************
14606
14607  052374  012777  052436  177344  ESR23:  MOV     #ESR24,@DEVECT
14608  052402  106427  000200                  MTPS    #200                    ;CPU AT LEVEL FOUR
14609  052406  012777  000001  177340          MOV     #1,@CSR1                ;TRY TO CAUSE AN INTERRUPT
14610  052414  012777  000007  177334          MOV     #7,@CSR2                ;AT LEVEL FIVE
14611  052422  000240                          NOP
14612  052424  012777  000006  177324          MOV     #6,@CSR2                ;CLRS GO, SETS DONE
14613  052432  000240                          NOP
14614  052434  104000                          EMT                             ;INTR DID NOT HAPPEN
14615  052436  012777  052472  177302  ESR24:  MOV     #ESR27,@DEVECT          ;ALTER INTR VECTOR
14616  052444  106427  000240                  MTPS    #240                    ;CHANGE LEVEL TO FIVE
14617  052450  012777  000007  177300          MOV     #7,@CSR2
14618  052456  000240                          NOP
14619  052460  012777  000000  177270          MOV     #0,@CSR2
14620  052466  000240                          NOP
14621  052470  000401                          BR      ESR28
14622  052472  104000                  ESR27:  EMT                             ;IF HERE, AN ERROR
14623                                   ;*****************************************************************
14624                                   ;       TRY INTERRUPT AT LEVEL SIX
14625                                   ;*****************************************************************
14626  052474  012777  052532  177244  ESR28:  MOV     #ESR29,@DEVECT
14627  052502  012777  000001  177244          MOV     #1,@CSR1                ;INTR RQST BITS TO 6
14628  052510  012777  000013  177240          MOV     #13,@CSR2
14629  052516  000240                          NOP
14630  052520  012777  000012  177230          MOV     #12,@CSR2               ;INTR SHOULD BE HONORED
14631  052526  000240                          NOP
14632  052530  104000                          EMT                             ;OTHERWISE AN ERROR
14633  052532  012777  052566  177206  ESR29:  MOV     #ESR31,@DEVECT          ;ALTER CPU INTR VECTOR
14634  052540  106427  000300                  MTPS    #300
14635  052544  012777  000013  177204          MOV     #13,@CSR2               ;SETS GO, CLRS DONE
14636  052552  000240                          NOP
14637  052554  012777  000000  177174          MOV     #0,@CSR2
14638  052562  000240                          NOP
14639  052564  000401                          BR      ESR30                   ;INTR SHOULD NOT BE HONORED
14640  052566  104000                  ESR31:  EMT                             ;BUT IF HERE, IT WAS
14641
14642                                   ;*****************************************************************
14643                                   ;       GENERATE AN INTERRUPT AT LEVEL SEVEN
14644                                   ;*****************************************************************
14645  052570  012777  052626  177150  ESR30:  MOV     #ESR33,@DEVECT
14646  052576  012777  000001  177150          MOV     #1,@CSR1                ;CPU AT LEVEL 6, Q22 AT 7
14647  052604  012777  000033  177144          MOV     #33,@CSR2
14648  052612  000240                          NOP
14649  052614  012777  000032  177134          MOV     #32,@CSR2
14650  052622  000240                          NOP
14651  052624  104000                          EMT
14652  052626  012777  052702  177112  ESR33:  MOV     #ESR35,@DEVECT          ;MODIFY INTR VECTOR IN CPU
14653  052634  106427  000340                  MTPS    #340
14654  052640  012777  000033  177110          MOV     #33,@CSR2
14655  052646  000240                          NOP
14656  052650  012777  000032  177100          MOV     #32,@CSR2
14657  052656  000240                          NOP
14658  052660  012777  000001  177070          MOV     #1,@CSR2                ;THIS WILL REMOVE THE
14659  052666  012777  000000  177062          MOV     #0,@CSR2                ;PENDING INTERRUPT
14660  052674  106427  000004                  MTPS    #4                      ;RESTORE THE PSW
```

```
14661  052700  000401                        BR      Q22TS1                  ;NADA SHOULD INTERRUPT
14662  052702  104000                ESR35:  EMT                             ;ERROR IF IT DOES
14663
14664                                ;*************************************************************************
  (2)                                ;TEST 425       DMA  DATO TRANSFER
  (3)                                ;*************************************************************************
  (2)  052704                        TS425:
14665
14666                                ;*************************************************************
14667                                ;
14668                                ;       THIS TEST WILL SET UP AN ACTUAL DMA TRANSFER
14669                                ;
14670                                ;*************************************************************
14671
14672  052704  012777  065432  177052  Q22TS1: MOV     #65432,@DATA
14673  052712  012777  001601  177034          MOV     #1601,@CSR1
14674  052720  005077  177032                  CLR     @CSR2                   ;NO INTERRUPT BITS
14675  052724  012777  177776  177030          MOV     #177776,@WC                     ;TWO WORDS
14676  052732  012777  050752  177020          MOV     #CNTR,@BA
14677  052740  005067  176006                  CLR     CNTR
14678  052744  005067  176004                  CLR     CNTR+2                          ;CLEAR THESE FIRST
14679  052750  012777  000001  177014          MOV     #1,@SIMGOA                      ;SIMULT. GO BIT
14680  052756  032777  000200  176772  ESR26:  BIT     #BIT7,@CSR2             ;WAIT FOR DONE
14681  052764  001001                          BNE     ESR25
14682  052766  000773                          BR      ESR26
14683  052770  022767  065432  175754  ESR25:  CMP     #65432,CNTR
14684  052776  001004                          BNE     ESR32
14685  053000  022767  065432  175746          CMP     #65432,CNTR+2                   ;GOOD DATA
14686  053006  001401                          BEQ     Q22TS2
14687  053010  104000                ESR32:  EMT                             ;NO, BAD DATA
14688
14689
14690
14691                                ;*************************************************************************
  (2)                                ;TEST 426       TEST PWR OK LOGIC
  (3)                                ;*************************************************************************
  (2)  053012                        TS426:
14692
14693                                ;*************************************************************
14694                                ;
14695                                ;       THIS TEST WILL DETERMINE IF THE CPU WILL RESPOND
14696                                ;       TO THE BPOK LINE BEING ALTERED. THE Q22BE
14697                                ;       WILL BE USED TO CONTROL THAT LINE ON THE Q BUS.
14698                                ;
14699                                ;*************************************************************
14700
14701  053012  000244                Q22TS2: CLZ
14702  053014  032777  000400  146554          BIT     #BIT8,@SWR              ;IS A Q BUS EXCER THERE
14703  053022  001002                          BNE     ESR20                   ;NO, IT'S A Q22
14704  053024  000167  000464                  JMP     @DVTST                  ;YES, THEN BYPASS THESE TESTS
14705
14706  053030  016767  124770  175706  ESR20:  MOV     24,TMP
14707  053036  016767  124764  175702          MOV     26,TMP1                 ;SAVE FOR LATER
14708  053044  012767  053116  124752          MOV     #ESR12,24
14709  053052  012767  000340  124746          MOV     #340,26         ;SET UP NEW VECTOR
14710  053060  010667  175666                  MOV     SP,CNTR                 ;SAVE STACK POINTER
```

M2



```
14711  053064  012706  000420                MOV     #420,SP                 ;MOVE SP WAY DOWN
14712  053070  012767  053130  124706        MOV     #ESR13,4                ;WHERE A STK OVFL WLD GO
14713  053076  012767  000340  124702        MOV     #340,6
14714  053104  012777  000040  176644        MOV     #40,@CSR2               ;SET BIT 5 IN Q22BE CSR2
14715  053112  000240                        NOP                             ;WILL PULL BPOK H LOW
14716  053114  104000                        EMT                             ;DIDNT CAUSE CPU TO GO
14717  053116  012777  000000  176632  ESR12: MOV     #0,@CSR2                ;THRU LOC 24
14718  053124  000240                        NOP                             ;BPOK H WILL GO HIGH
14719  053126  000401                        BR      ESR14
14720  053130  104000                ESR13:  EMT                             ;IF HERE, AN ERROR BY GOING
14721                                                                                  ;THRU LOC 4
14722  053132  016706  175614        ESR14:  MOV     CNTR,SP         ;RESTORE SOME STUFF
14723  053136  016767  175602  124660        MOV     TMP,24
14724  053144  016767  175576  124654        MOV     TMP1,26
14725  053152  000167  000000                JMP     Q22TS3                  ;GO TO NEXT TEST
14726
14727
14728
14729                                ;*****************************************************************************************
  (2)                               ;TEST 427       TEST INDIVIDUAL EXTENDED ADRS BITS
  (3)                               ;*****************************************************************************************
  (2)  053156                       TS427:
14730
14731                               ;*************************************************************
14732                               ;
14733                               ;       THIS TEST WILL UTILIZE THE Q22BE LATENCY CNTR
14734                               ;       TO CAPTURE THE EXTENDED ADDRESS BITS ON THE Q BUS.
14735                               ;       THE Q22BE LATENCY COUNTER BITS 15-12 CORRESPOND
14736                               ;       TO ADDRESS BITS 21-18.
14737                               ;
14738                               ;*************************************************************
14739
14740  053156  016767  124622  175560  Q22TS3: MOV     4,TMP
14741  053164  016767  124616  175554          MOV     6,TMP1                  ;STORE FOR LATER
14742  053172  012767  053270  124604          MOV     #ESR15,4
14743  053200  012767  000340  124600          MOV     #340,6                  ;NEW VECTOR AND PSW
14744
14745  053206  017700  176554                 MOV     @LATCNT,R0                      ;READ IT TO CLR IT
14746  053212  012767  000020  117276          MOV     #BIT4,SR3               ;  22 BIT ADRSNG
14747  053220  012767  010000  117122          MOV     #10000,KIPAR4                   ;SET FOR ADRS BIT 18
14748  053226  012767  010000  175520          MOV     #10000,CNTR+2
14749  053234  012767  000000  175510          MOV     #0,CNTR                 ;ZERO THESE
14750  053242  012767  000001  124322          MOV     #BIT0,SR0                       ;TURN ON MEM MANG
14751  053250  026727  175476  000017  ESR17:  CMP     CNTR,#17                ;FINISHED ?
14752  053256  001433                  BEQ     ESR16
14753  053260  105037  100000                  CLRB    @#100000                ;IF THERE IS NO MEMORY HERE
14754  053264  000240                  NOP                             ;SHOULD HAVE TIMED OUT
14755  053266  000240                  NOP                             ;WE ARE IN BIG TROUBLE
14756
14757  053270  017700  176472          ESR15:  MOV     @LATCNT,R0              ;READ LATENCY COUNTER
14758  053274  042700  007777                  BIC     #7777,R0                ;CLR DONT CARES
14759  053300  026700  175450                  CMP     CNTR+2,R0               ;EXPECTED ?
14760  053304  001407                  BEQ     ESR18                   ;EQUALS RECVD
14761  053306  012767  000000  124256          MOV     #0,SR0                  ;TURN OFF MM AND 22 BITS
14762  053314  012767  000000  117174          MOV     #0,SR3
14763  053322  104000                  EMT                             ;DATA NOT GOOD
```

```
14764  053324  005267  175422          ESR18:  INC     CNTR
14765  053330  062767  010000  117012          ADD     #10000,KIPAR4
14766  053336  062767  010000  175410          ADD     #10000,CNTR+2           ;INCREASE ADRS BITS BY ONE
14767  053344  000741                          BR      ESR17                   ;CONTINUE
14768
14769  053346  016767  175372  124430  ESR16:  MOV     TMP,4
14770  053354  016767  175366  124424          MOV     TMP1,6                  ;RESTORE THIS VECTOR
14771  053362  012767  000000  124202          MOV     #0,SR0                  ;TURN OFF MM AND 22 BITS
14772  053370  012767  000000  117120          MOV     #0,SR3
14773  053376  000167  000112                  JMP     BDVTST
14774
14775
14776                                  ;*****************************************************************************
14777
14778  053402  004767  060022          ERRORB: JSR     PC,ABORT                ;ARE WE UNDER UFD ?
14779  053406  012737  000011  001002          MOV     #11,@#$FATAL
14780  053414  012767  000001  125356          MOV     #1,$MSGTY
14781  053422  032737  000001  001020          BIT     #1,@#$ENV
14782  053430  001004                          BNE     Q22HLT
14783  053432  012700  053444                  MOV     #Q22MSG,R0
14784  053436  004767  057722                  JSR     PC,TYPE
14785  053442  000777                  Q22HLT: BR      .
14786  053444  040506  046111  051125  Q22MSG: .ASCIZ   /FAILURE DURING Q BUS EXERCISER TESTS/<12><15>
       053452  020105  052504  044522
       053460  043516  050440  041040
       053466  051525  042440  040130
       053474  041522  051511  051105
       053502  052040  051505  051524
       053510  006412     000
14787          053514                  .EVEN
14788
14789                                  ;*****************************************************************************
14790
14791
14792
14793                                  ;*****************************************************************
14794                                  ;
14795                                  ;       BDV TESTS
14796                                  ;
14797                                  ;
14798                                  ;*****************************************************************
14799
14800                                  ;*************************************************************************************************
  (2)                                  ;TEST 430       TEST FOR FUNCTIONALITY OF R/W REGISTER
  (3)                                  ;*************************************************************************************************
  (2)  053514                          TS430:
14801
14802
14803
14804  053514                          BDVTST:
14805  053514  012767  000017  124002          MOV     #17,LSREG               ;TURN OFF THE 4 LEDS
14806  053522  012737  000012  001004          MOV     #12,@#$TESTN            ;TST NUMBER FOR APT
14807  053530  012737  054502  000030          MOV     #ERRORD,@#30            ;ERROR TRAP
14808  053536  000244                          CLZ
14809  053540  032777  002000  146030          BIT     #BIT10,@SWR             ;WANT TO TEST E102 ?
14810  053546  001410                          BEQ     ESR10                   ;NO
```

```
14811  053550  016767  123750  175166          MOV     LSREG,TMP               ;RD THIS TO GET SWITCHES
14812  053556  026727  175162  000317          CMP     TMP,#317                ;SHOULD BE THIS VALUE
14813  053564  001401                          BEQ     ESR10                   ;EQUAL
14814  053566  104000                          EMT
14815  053570  012737  054402  000030  ESR10:  MOV     #ERRORC,@#30            ;ERROR TRAP
14816  053576  005067  123720                  CLR     RWREG           ;THE NEXT FEW INSTRUCTIONS
14817  053602  016701  123714                  MOV     RWREG,R1                ;WILL TEST THE R/W ABILITY
14818  053606  020127  000000                  CMP     R1,#0                   ;OF THE R/W MAINT. REGISTER
14819  053612  001401                          BEQ     ESR40
14820  053614  104000                          EMT
14821  053616  012767  177777  123676  ESR40:  MOV     #-1,RWREG
14822  053624  016701  123672                  MOV     RWREG,R1
14823  053630  022701  177777                  CMP     #177777,R1
14824  053634  001401                          BEQ     ESR41
14825  053636  104000                          EMT
14826  053640  012767  125252  123654  ESR41:  MOV     #125252,RWREG
14827  053646  016701  123650                  MOV     RWREG,R1
14828  053652  020127  125252                  CMP     R1,#125252
14829  053656  001401                          BEQ     ESR42
14830  053660  104000                          EMT
14831  053662  105067  123634          ESR42:  CLRB    RWREG                   ;BYTES
14832  053666  016701  123630                  MOV     RWREG,R1
14833  053672  020127  125000                  CMP     R1,#125000
14834  053676  001401                          BEQ     ESR43
14835  053700  104000                          EMT
14836  053702  000367  123614          ESR43:  SWAB    RWREG
14837  053706  016701  123610                  MOV     RWREG,R1
14838  053712  020127  000252                  CMP     R1,#252
14839  053716  001401                          BEQ     ESR44
14840  053720  104000                          EMT
14841  053722  012767  052525  123572  ESR44:  MOV     #52525,RWREG
14842  053730  016701  123566                  MOV     RWREG,R1
14843  053734  020127  052525                  CMP     R1,#52525
14844  053740  001401                          BEQ     ESR45
14845  053742  104000                          EMT
14846  053744                          ESR45:
14847  053744  105067  123553                  CLRB    RWREG+1
14848  053750  016701  123546                  MOV     RWREG,R1
14849  053754  020127  000125                  CMP     R1,#125
14850  053760  001401                          BEQ     ESR46
14851  053762  104000                          EMT
14852  053764  000367  123532          ESR46:  SWAB    RWREG
14853  053770  016701  123526                  MOV     RWREG,R1
14854  053774  020127  052400                  CMP     R1,#52400
14855  054000  001401                          BEQ     ESR47
14856  054002  104000                          EMT
14857  054004  005067  123512          ESR47:  CLR     RWREG
14858  054010  052767  100000  123504          BIS     #BIT15,RWREG            ;SET 15
14859  054016  016701  123500                  MOV     RWREG,R1
14860  054022  020167  123474          ROTLP1: CMP     R1,RWREG                ;ARE THEY THE SAME
14861  054026  001401                          BEQ     ESR48
14862  054030  104000                          EMT
14863  054032  006001                  ESR48:  ROR     R1                      ;NEXT BIT TO THE RIGHT
14864  054034  001403                          BEQ     ESR49
14865  054036  006067  123460                  ROR     RWREG
14866  054042  000767                          BR      ROTLP1                  ;CONTINUE TESTING
```

```
14867  054044  012767  177777  123450  ESR49:  MOV     #-1,RWREG
14868  054052  042767  100000  123442          BIC     #BIT15,RWREG
14869  054060  016701  123436                  MOV     RWREG,R1
14870  054064  026701  123432          ROTLP2: CMP     RWREG,R1                ;SAME ?
14871  054070  001401                          BEQ     ESR50                   ;YES
14872  054072  104000                          EMT
14873  054074  000261          ESR50:  SEC                                     ;SET THE C BIT
14874  054076  006067  123420                  ROR     RWREG
14875  054102  006001                          ROR     R1
14876  054104  020127  077777                  CMP     R1,#77777
14877  054110  001365                          BNE     ROTLP2                  ;NOT FINISHED YET
14878  054112  000431                          BR      BDVTS2
14879
14880
14881                          ;*****************************************************************************
  (2)                          ;TEST 431       ROM CHECKSUM TEST
  (3)                          ;*****************************************************************************
  (2)  054114                  TS431:
14882
14883                          ;*****************************************************************
14884                          ;
14885                          ;       THE PREVIOUS TEST CHECKED OUT THE ROM R/W REGISTER
14886                          ;       NOW WE WILL TEST THE 8K DIAGNOSTIC ROM FOR THE
14887                          ;       CORRECT CHECKSUM AND CHECKWORD
14888                          ;
14889                          ;*****************************************************************
14890
14891
14892
14893                          ;*****************************************************************
14894                          ;
14895                          ;       DATA SECTION FOR THE NEXT TEST
14896                          ;
14897                          ;*****************************************************************
14898  054114  000000                  VRTPCR: .WORD   0                       ;VIRTUAL PAGE CONTROL REGISTER
14899  054116  000000                  BCF:    .WORD   0
14900  054120  000000                  COUNTR: .WORD   0
14901  054122  000001                  ANSR:   .WORD   1
14902  054124  000000                  RFLAG:  .WORD   0
14903  054126  000000                  EXPSUM: .WORD   0
14904  054130  000000                  ACTSUM: .WORD   0
14905
14906                          ;*************************************************************
14907                          ;FUNCTIONAL DESCRIPTION:
14908                          ;SUBROUTINE TO COMPUTE A CHECKSUM IN A ROM/EPROM
14909                          ;INPUT: CONTENTS OF BCF
14910                          ;IMPLICIT INPUTS: CONTENTS OF PCR
14911                          ;OUTPUT: A CHECKSUM VALUE STORED IN LOCATION ACTSUM
14912                          ;CALLING SEQUENCE: JSR PC,CHKSUM
14913                          ;*************************************************************
14914
14915  054132  012701  173776  CHKSUM: MOV     #173776,R1              ;STORE THE HIGHEST ADDRESS IN THE ROM
14916  054136  066701  177754          ADD     BCF,R1                  ;FOR EITHER LOW OR HIGH BYTES
14917  054142  005067  177762          CLR     ACTSUM                  ;CLEAR LOCATION WHICH WILL HOLD THE CHECKSUM
14918  054146  012702  173000          MOV     #173000,R2              ;COMPUTE THE LOWEST ADDRESS IN THE ROM
14919  054152  066702  177740          ADD     BCF,R2                  ;WHERE THE DATA WILL START
```

```
14920  054156  111204                  1$:     MOVB    (R2),R4                 ;GET DATA IN BYTES
14921  054160  060467  177744                  ADD     R4,ACTSUM               ;ADD CONTENTS OF EACH LOCATION TO THE CHECKSUM
14922  054164  062702  000002                  ADD     #2,R2                   ;ADJUST ADDRESS
14923  054170  020201                          CMP     R2,R1                   ;COMPARE CURRENT ADDRESS WITH HIGHEST ADDRESS
14924  054172  002771                          BLT     1$                      ;BR IF LESS THAN
14925  054174  000207                          RTS     PC                      ;RETURN
14926
14927
14928                                  ;*****************************************************************
14929                                  ;TEST TO PERFORM CHECKSUM AND CHECKWORD VERIFICATION ON THE 8K
14930                                  ;OF DIAGNOSTIC ROM. IN UNATTENDED MODE, THE ROM WILL BE ADDRESSED
14931                                  ;FROM 0-8K. IN STAND-ALONE MODE, THE OPERATOR MAY CHANGE THE
14932                                  ;ADDRESS BY RESPONDING TO QUESTIONS GENERATED ON THE FIRST PASS.
14933                                  ;*****************************************************************
14934
14935  054176  012767  000400  000172  BDVTS2: MOV     #400,DRLP               ;STORE STARTING ADDRESS
14936  054204  016767  000166  177702          MOV     DRLP,VRTPCR             ;SET UP PCR
14937  054212  016767  177676  123300          MOV     VRTPCR,PCR
14938  054220  012767  000040  177672          MOV     #40,COUNTR              ;SET NUMBER OF CHECKWORDS TO CHECK
14939  054226  012767  000001  177670          MOV     #1,RFLAG                ;INDICATE ROM
14940  054234  005067  177656          DLOOP:  CLR     BCF                     ;SIGNAL LOW BYTES ARE BEING CHECKED
14941  054240  122737  177777  173774          CMPB    #-1,@#173774            ;DOES THE ROM EXIST?
14942  054246  001001                          BNE     1$                      ;BR IF YES
14943  054250  104000                          EMT
14944  054252  004767  177654          1$:     JSR     PC,CHKSUM               ;COMPUTE THE ACTUAL CHECKSUM
14945  054256  113767  173776  177642          MOVB    @#173776,EXPSUM         ;GET THE STORED CHECKSUM
14946  054264  066767  177640  177634          ADD     ACTSUM,EXPSUM           ;ADD THE EXPECTED AND ACTUAL CHECKSUMS
14947  054272  105767  177630                  TSTB    EXPSUM                  ;BYTE RESULT = 0?
14948  054276  001401                          BEQ     2$                      ;BR IF YES
14949  054300  104000                          EMT
14950  054302  012767  000001  177606  2$:     MOV     #1,BCF                  ;SET BCF TO DENOTE HIGH BYTES
14951  054310  122737  177777  173775          CMPB    #-1,@#173775            ;DOES THE ROM EXIST?
14952  054316  001001                          BNE     3$                      ;BR IF YES
14953  054320  104000                          EMT
14954  054322  004767  177604          3$:     JSR     PC,CHKSUM               ;COMPUTE THE ACTUAL CHECKSUM
14955  054326  113767  173777  177572          MOVB    @#173777,EXPSUM         ;GET EXPECTED CHECKSUM
14956  054334  066767  177570  177564          ADD     ACTSUM,EXPSUM           ;ADD THE EXPECTED AND ACTUAL CHECKSUMS
14957  054342  105767  177560                  TSTB    EXPSUM                  ;BYTE RESULT = 0?
14958  054346  001401                          BEQ     4$                      ;BR IF YES
14959  054350  104000                          EMT
14960  054352  062767  001002  177534  4$:     ADD     #1002,VRTPCR            ;NEXT PAGE IN PCR
14961  054360  016767  177530  123132          MOV     VRTPCR,PCR
14962  054366  005367  177526                  DEC     COUNTR                  ;DECREMENT CHECKWORD COUNT
14963  054372  001320                          BNE     DLOOP                   ;LOOP UNTIL ALL 20 PAGES HAVE BEEN CHECKED
14964
14965                                  ;
14966                                  ; GO DO THE FOATING POINT TESTS
14967                                  ;
14968
14969  054374  000510                  COMPLE: BR      FPSTRT
14970
14971
14972
14973
14974  054376  000000                  DRLP:   .WORD   0
14975  054400  000000                  PNTR:   .WORD   0               ;WILL BE USED AS A POINTER
```

```
14976
14977                                   ;*****************************************************************
14978                                   ;
14979                                   ;       ERROR ROUTINE FOR THE BDV TESTING
14980                                   ;
14981                                   ;*****************************************************************
14982
14983   054402  004767  057022          ERRORC: JSR     PC,ABORT                        ;ARE WE UNDER UFD ?
14984   054406  012737  000012  001002          MOV     #12,@#$FATAL
14985   054414  012767  000001  124356          MOV     #1,$MSGTY
14986   054422  032737  000001  001020          BIT     #1,@#$ENV                       ;UNDER APT ?
14987   054430  001004                          BNE     BDVHLT
14988   054432  012700  054444                  MOV     #BDVMSG,R0
14989   054436  004767  056722                  JSR     PC,TYPE
14990   054442  000777                  BDVHLT: BR      .
14991
14992   054444  040506  046111  042105  BDVMSG: .ASCIZ  /FAILED DURING THE BDV TESTS/<12><15>
        054452  042040  051125  047111
        054460  020107  044124  020105
        054466  042102  020126  042524
        054474  052123  005123  000015
14993
14994
14995   054502  004767  056722          ERRORD: JSR     PC,ABORT                        ;ARE WE UNDER UFD ?
14996   054506  012737  000012  001002          MOV     #12,@#$FATAL
14997   054514  012767  000001  124256          MOV     #1,$MSGTY
14998   054522  032737  000001  001020          BIT     #1,@#$ENV
14999   054530  001344                          BNE     BDVHLT
15000   054532  012700  054544                  MOV     #SWMSG,R0
15001   054536  004767  056622                  JSR     PC,TYPE
15002   054542  000000                          HALT
15003   054544  044103  041505  020113  SWMSG:  .ASCIZ  /CHECK SWITCHES ON E102, RESTART AT 200/<12><15>
        054552  053523  052111  044103
        054560  051505  047440  020116
        054566  030505  031060  020054
        054574  042522  052123  051101
        054602  020124  052101  031040
        054610  030060  006412     000
15004           054616                  .EVEN
15005
15006                                   ;* ****************************************************************
15014
15015
15016
15023
15024
15031
15047
15076
15077           000244                  FPVECT=244
15078                                           .SBTTL  FPP REGISTER DEFINITIONS
15079           000000                  AC0     =%0
15080           000001                  AC1     =%1
15081           000002                  AC2     =%2
15082           000003                  AC3     =%3
15083           000004                  AC4     =%4
```

```
15084            000005                  AC5     =%5
15085            000006                  AC6     =%6
15086            000007                  AC7     =%7
15087
15088
15089  054616    012706  001000          FPSTRT: MOV     #STBOT,SP       ;SET UP STACK POINTER
15090  054622    000244                          CLZ
15091  054624    032777  000002  144744          BIT     #2,@SWR
15092  054632    001002                          BNE     1$
15093  054634    000167  047776                  JMP     SLU1ST
15094  054640    012737  124452  000030  1$:     MOV     #ERROR4,@#30    ;SETUP FOR CORRECT ERROR CALL
15095  054646    012737  000003  001004          MOV     #3,@#$TESTN     ;PUT TEST NUMBER IN MAILBOX
15096
15097
15108                                     ;*****************************************************************************
  (2)                                     ;TEST 432       LDFPS, STFPS AND DATA PATHS TEST
  (3)                                     ;*****************************************************************************
  (2)  054654                             TS432:
15109  054654    012700  177777                  MOV     #-1,R0                  ;INITIALIZE THE COUNT PATTERN.
15110  054660    012737  054732  000244          MOV     #AERR1,@#FPVECT         ;SET UP FOR UNABLE TO DECODE
15111  054666    012737  054732  000010          MOV     #AERR1,@#10             ;FPP INSTRUCTION TRAP TO 244 OR 10.
15112  054674    012737  054732  000004          MOV     #AERR1,@#ERRVECT        ;IF EITHER INSTRUCTION
15113                                                                           ;FAILS TO GO THROUGH THE
15114                                                                           ;CORRECT SRC OR DST MODE AN
15115                                                                           ;ODD ADDRESS TRAP WILL OCCUR.
15116  054702                             A1:
15117  054702    010004                  A11:    MOV     R0,R4
15118  054704    042704  030020                  BIC     #30020,R4
15119  054710    170104                          LDFPS   R4                      ;TEST INSTRUCTION.
15120
15121  054712    012701  177777                  MOV     #-1,R1
15122  054716    170201                  A12:    STFPS   R1                      ;TEST INSTRUCTION.
15123  054720    010004                          MOV     R0,R4                   ;MASK OFF UNSETTABLE BITS.
15124  054722    042704  030020                  BIC     #30020,R4
15125  054726    020401                          CMP     R4,R1                   ;COMPARE DATA EXPECTED WITH
15126                                                                           ;THE DATA READ.
15127  054730    001401                          BEQ     A2
  (1)  054732                             AERR1:
  (2)  054732    104000                          EMT                     ;
15128
15129  054734    012700  000001          A2:     MOV     #1,R0                   ;NEXT PATTERN WILL BE ALL ZERO
15130  054740    077020                          SOB     R0,A1                   ;DECREMENT COUNT PATTERN
15131  054742                             ADONE:
  (1)  054742    004767  047604                  JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                     ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                     ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                     ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                     ;THE USER TYPED CONTROL G?).
15132
15133
15139                                     ;*****************************************************************************
  (2)                                     ;TEST 433       CFCC TEST
  (3)                                     ;*****************************************************************************
  (2)  054746                             TS433:
15140  054746    012700  000017                  MOV     #17,R0                  ;R0 CONTAINS TO TEST PATTERN.
15141
```

```
15142  054752                          B1:
15143  054752  170100                          LDFPS   R0                      ;LOAD THE TEST PATTERN
15144
15145  054754                          B2:
15146  054754  170000                          CFCC                            ;COPY CONDITION CODES.
15147
15148  054756  013703  177776                  MOV     @#PSW,R3                ;SEE IF PATTERN TRANSFERED.
15149  054762  042703  177760                  BIC     #177760,R3
15150  054766  020003                          CMP     R0,R3
15151  054770  001401                          BEQ     B3
  (2)  054772  104000                          EMT                     ;
15152  054774  077012                  B3:     SOB     R0,B1
15153  054776                          BDONE:
  (1)  054776  004767  047550                  JSR     PC,..RSET               ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                          ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                          ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                          ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                          ;THE USER TYPED CONTROL G?).
15154
15165                                   ;*************************************************************************************
  (2)                                   ;TEST 434       SETF, SETD, SETI AND SETL TEST
  (3)                                   ;*************************************************************************************
  (2)  055002                           TS434:
15166  055002  005000                          CLR     R0
15167
15168  055004  170100                          LDFPS   R0                      ;CLEAR THE FPS.
15169  055006  170001                  C15:    SETF                            ;TEST INSTRUCTION.
15170
15171  055010  170201                          STFPS   R1                      ;GET RESULT.
15172  055012  005002                          CLR     R2
15173  055014  020201                          CMP     R2,R1                   ;DID AN ERROR OCCUR?
15174  055016  001401                          BEQ     C2
  (2)  055020  104000                          EMT                     ;
15175  055022  012700  147757          C2:     MOV     #147757,R0
15176
15177  055026  170100                          LDFPS   R0                      ;PUT 147757 IS FPS
15178  055030  170001                  C25:    SETF                            ;CLEAR FD BIT.
15179
15180  055032  170201                          STFPS   R1                      ;GET RESULT
15181  055034  012702  147557                  MOV     #147557,R2
15182  055040  020102                          CMP     R1,R2                   ;RESULT CORRECT.
15183  055042  001401                          BEQ     C3
  (2)  055044  104000                          EMT                     ;
15184  055046  012700  147757          C3:     MOV     #147757,R0
15185
15186  055052  170100                          LDFPS   R0                      ;LOAD 147757 INTO FPS.
15187  055054  170011                  C35:    SETD                            ;SETD FD BIT.
15188
15189  055056  170201                          STFPS   R1
15190  055060  012702  147757                  MOV     #147757,R2
15191  055064  020102                          CMP     R1,R2                   ;RESULT CORRECT?
15192  055066  001401                          BEQ     C4
  (2)  055070  104000                          EMT                     ;
15193  055072  005000                  C4:     CLR     R0
15194  055074  170100                          LDFPS   R0                      ;CLEAR FPS.
15195  055076  170011                  C45:    SETD                            ;SET FD BIT.
```

H3



```
15196
15197  055100  170201                      STFPS   R1                      ;GET RESULT.
15198  055102  012702  000200              MOV     #200,R2
15199  055106  020102                      CMP     R1,R2                   ;RESULT CORRECT?
15200  055110  001401                      BEQ     C5
  (2)  055112  104000                      EMT                     ;
15201  055114  005000              C5:     CLR     R0
15202
15203  055116  170100                      LDFPS   R0                      ;CLEAR FPS
15204  055120  170002              C55:    SETI                            ;CLEAR FL BIT.
15205
15206  055122  170201                      STFPS   R1                      ;GET RESULT.
15207  055124  005002                      CLR     R2
15208  055126  020201                      CMP     R2,R1                   ;RESULT CORRECT?
15209  055130  001401                      BEQ     C6
  (2)  055132  104000                      EMT                     ;
15210  055134  012700  147757      C6:     MOV     #147757,R0
15211  055140  170100                      LDFPS   R0                      ;PUT 147757 INTO FPS
15212  055142  170002              C65:    SETI                            ;CLEAR FL BIT.
15213
15214  055144  170201                      STFPS   R1                      ;GET THE RESULT.
15215  055146  012702  147657              MOV     #147657,R2
15216  055152  020102                      CMP     R1,R2                   ;RESULT CORRECT?
15217  055154  001401                      BEQ     C7
  (2)  055156  104000                      EMT                     ;
15218  055160  012700  147757      C7:     MOV     #147757,R0
15219  055164  170100                      LDFPS   R0                      ;SET FPS TO 147757.
15220  055166  170012              C75:    SETL                            ;SET FL BIT.
15221
15222  055170  170201                      STFPS   R1                      ;GET THE RESULT.
15223  055172  012702  147757              MOV     #147757,R2
15224  055176  020102                      CMP     R1,R2                   ;RESULT CORRECT?
15225  055200  001401                      BEQ     C8
  (2)  055202  104000                      EMT                     ;
15226  055204  005000              C8:     CLR     R0
15227  055206  170100                      LDFPS   R0                      ;CLEAR FPS.
15228  055210  170012              C85:    SETL                            ;SET FL BIT.
15229
15230  055212  170201                      STFPS   R1
15231  055214  012702  000100              MOV     #100,R2
15232  055220  020102                      CMP     R1,R2                   ;RESULT CORRECT.
15233  055222  001401                      BEQ     CDONE
  (2)  055224  104000                      EMT                     ;
15234  055226                      CDONE:
  (1)  055226  004767  047320              JSR     PC,.RSET                ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
15235
15236
15256                             ;*************************************************************************
  (2)                             ;TEST 435       ILLEGAL FPP OP CODES AND STST TEST
  (3)                             ;*************************************************************************
  (2)  055232                     TS435:
15257  055232  012705  170003              MOV     #170003,R5              ;INITIAL OP CODE.
```

```
15258  055236  012737  055276  000004          MOV     #DERR2,@#ERRVECT
15259  055244  012737  055330  000244          MOV     #DERR1,@#FPVECT
15260
15261  055252  005000                  D1:     CLR     R0
15262  055254  170100                          LDFPS   R0                      ;CLEAR FPS.
15263  055256  005002                          CLR     R2
15264  055260  010537  055264                  MOV     R5,@#D2                 ;SET UP THE ILLEGAL INSTRUCTION.
15265  055264  000000                  D2:     .WORD   0
15266  055266  170000                  D3:     CFCC
15267  055270  005202                          INC     R2
15268  055272  005202                  D4:     INC     R2
15269
15270  055274  170201                          STFPS   R1                      ;REPORT FAILURE. DID NOT TRAP.
15271  055276                          DERR2:
  (2)  055276  104000                          EMT                     ;
15272  055300  022705  170010          D5:     CMP     #170010,R5              ;COMPUTE NEXT OP CODE
15273  055304  001003                          BNE     D6
15274  055306  012705  170013                  MOV     #170013,R5
15275  055312  000757                          BR      D1
15276
15277  055314  022705  170077          D6:     CMP     #170077,R5
15278  055320  001001                          BNE     D7
15279  055322  000424                          BR      DDONE
15280  055324  005205                  D7:     INC     R5
15281  055326  000751                          BR      D1
15282
15283  055330  022716  055266          DERR1:  CMP     #D3,(SP)                ;DID TRAP OCCUR ON TEST INSTRUCTION?
15284  055334  001401                          BEQ     1$
  (2)  055336  104000                          EMT                     ;
15285  055340  022626                  1$:     CMP     (SP)+,(SP)+
15286  055342  170201                          STFPS   R1                      ;GET THE FPS AND SEE IF IT IS
15287  055344  022701  100000                  CMP     #100000,R1              ;SET CORRECTLY.
15288  055350  001401                          BEQ     3$
  (2)  055352  104000                          EMT                     ;
15289  055354  012704  000001          3$:     MOV     #1,R4
15290  055360  170304                  D8:     STST    R4                              ;GET THE FEC CODE. NOTE THAT
15291                                                                                  ;IF THE DESTINATION MODE IS
15292                                                                                  ;IMPROPERLY DECODED AN ODD
15293                                                                                  ;ADDRESS TRAP TO 4 SHOULD OCCUR.
15294  055362  022704  000002                  CMP     #2,R4                   ;WAS FEC CORRECT?
15295  055366  001001                          BNE     D9
15296  055370  000743                          BR      D5
15297
15298  055372                          D9:                                     ;REPORT STST FAILURE
15299  055372  104000                          EMT                     ;
15300  055374                          DDONE:
  (1)  055374  004767  047152                  JSR     PC,..RSET               ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                           ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                           ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                           ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                           ;THE USER TYPED CONTROL G?).
15301
15309                           ;*****************************************************************************
  (2)                           ;TEST 436       FID, INTERRUPT DISABLE, BIT TEST
  (3)                           ;*****************************************************************************
  (2)  055400                   TS436:
```

J3



```
15310  055400  012737  055440  000244          MOV     @EERR0,@@FPVECT         ;SETUP FOR THE INTERRUPT.
15311
15312  055406  012700  040000          E1:     MOV     @40000,R0
15313  055412  170100                          LDFPS   R0                      ;SET FID.
15314  055414  170020                  E3:     .WORD   170020                  ;ILLEGAL FPP INSTRUCTION.
15315  055416  170000                  E4:     CFCC
15316
15317  055420  170201                          STFPS   R1                      ;SEE IF ERROR WAS DETECTED.
15318  055422  022701  140000                  CMP     @140000,R1
15319  055426  001004                          BNE     EERR0
15320
15321  055430  170304                          STST    R4                      ;SEE IF FEC=2
15322  055432  022704  000002                  CMP     @2,R4
15323  055436  001401                          BEQ     EDONE
  (1)  055440                          EERR0:
  (2)  055440  104000                          EMT                     ;
15324  055442                         EDONE:
  (1)  055442  004767  047104                  JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
15325
15326
15339                            ;*************************************************************************************
  (2)                            ;TEST 437       LDD AND STD, WITH SRC AND DST MODE 1, TEST
  (3)                            ;*************************************************************************************
  (2)  055446                    TS437:
15340
15341  055446  005000                  CLR     R0
15342  055450  170100                  LDFPS   R0
15343  055452  170011                  SETD
15344  055454  012701  055716          MOV     #FDATI0,R1              ;SET UP THE LOAD DATA.
15345  055460  012702  055762          MOV     #FXDAT0,R2
15346  055464  012703  000010          MOV     #10,R3
15347
15348  055470  012221          F2:     MOV     (R2)+,(R1)+
15349  055472  077302                  SOB     R3,F2
15350
15351  055474  012700  055726          MOV     #FDATI4,R0              ;SETUP R0 FOR THE LDD (R0),AC0.
15352  055500  012737  055714  000004  MOV     #FERR20,@@ERRVECT       ;IF THE SRC FLOWS FAIL THEN
15353                                                                   ;AN ODD ADDRESS MAY OCCUR.
15354  055506  005003                  CLR     R3
15355
15356  055510  172410          F3:     LDD     (R0),AC0
15357  055512  005203          F4:     INC     R3
15358  055514  005203                  INC     R3
15359
15360  055516  020027  055726          CMP     R0,#FDATI4              ;WAS R0 AFFECTED?
15361  055522  001401                  BEQ     F5
  (2)  055524  104000                  EMT                     ;
15362  055526  020327  000002  F5:     CMP     R3,#2                   ;SEE IF THE PC WAS ADVERSELY
15363  055532  001401                  BEQ     1$
  (2)  055534  104000                  EMT                     ;
15364  055536  012701  055716  1$:     MOV     #FDATI0,R1              ;MAKE SURE THE SOURCE DATA WAS
15365  055542  012702  055762          MOV     #FXDAT0,R2              ;NOT AFFECTED.
```

```
15366  055546  012703  000010          MOV     #10,R3
15367  055552  022122          2$:     CMP     (R1)+,(R2)+
15368  055554  001401                  BEQ     3$
  (2)  055556  104000                  EMT                             ;
15369  055560  077304          3$:     SOB     R3,2$
15370
15371  055562  170201                  STFPS   R1                      ;MAKE SURE THE FPS IS CORRECT.
15372  055564  022701  000200          CMP     #200,R1
15373  055570  001401                  BEQ     F6
  (2)  055572  104000                  EMT                             ;
15374  055574  012703  177777  F6:     MOV     #-1,R3
15375  055600  012704  000010          MOV     #10,R4
15376  055604  012705  055740          MOV     #FDAT00,R5              ;SET UP THE OUTPUT DATA BUFFER.
15377  055610  010325          F7:     MOV     R3,(R5)+
15378  055612  077402                  SOB     R4,F7
15379
15380  055614  012700  055750          MOV     #FDAT04,R0              ;SET UP R0 FOR DST MODE 1 REG 0.
15381  055620  012737  055714  000004  MOV     #FERR20,@#ERRVECT       ;IF THE DST FLOWS FAIL AN ODD
15382                                                                  ;ADDRESS COULD OCCUR.
15383  055626  005003                  CLR     R3
15384
15385  055630  174010          F10:    STD     AC0,(R0)                ;TEST INSTRUCTION.
15386  055632  005203          F11:    INC     R3
15387  055634  005203                  INC     R3
15388
15389  055636  020027  055750          CMP     R0,#FDAT04              ;WAS R0 MODIFIED?
15390  055642  001401                  BEQ     F12
  (2)  055644  104000                  EMT                             ;
15391  055646  020327  000002  F12:    CMP     R3,#2                   ;WAS THE PC AFFECTED CORRECTLY?
15392  055652  001401                  BEQ     F135
  (2)  055654  104000                  EMT                             ;
15393  055656  012701  055740  F135:   MOV     #FDAT00,R1
15394  055662  012702  055762          MOV     #FXDAT0,R2
15395  055666  012703  000010          MOV     #10,R3                  ;SETUP LOOP COUNT
15396  055672  022122          F13:    CMP     (R1)+,(R2)+             ;WAS DATA OUTPUT CORRECTLY
15397  055674  001401                  BEQ     F14
  (2)  055676  104000                  EMT                             ;
15398  055700  077304          F14:    SOB     R3,F13                  ;SUBTRACT 1 FROM LOOP COUNT AND LOOP IF NOT ZERO
15399  055702  005001          F22:    CLR     R1
15400  055704  170201                  STFPS   R1                              ;MAKE SURE FPS IS CORRECT.
15401  055706  022701  000200          CMP     #200,R1
15402  055712  001433                  BEQ     FDONE
  (1)  055714                  FERR20:
  (2)  055714  104000                  EMT                             ;
15403
15404  055716  177777          FDATI0: -1
15405  055720  177777          FDATI1: -1
15406  055722  177777          FDATI2: -1
15407  055724  177777          FDATI3: -1
15408  055726  177777          FDATI4: -1
15409  055730  177777          FDATI5: -1
15410  055732  177777          FDATI6: -1
15411  055734  177777          FDATI7: -1
15412  055736  177777                  -1
15413  055740  177777          FDAT00: -1
15414  055742  177777          FDAT01: -1
```

```
15415  055744  177777                  FDAT02: -1
15416  055746  177777                  FDAT03: -1
15417  055750  177777                  FDAT04: -1
15418  055752  177777                  FDAT05: -1
15419  055754  177777                  FDAT06: -1
15420  055756  177777                  FDAT07: -1
15421  055760  177777                          -1
15422  055762  177777                  FXDAT0: -1
15423  055764  177777                  FXDAT1: -1
15424  055766  177777                  FXDAT2: -1
15425  055770  177777                  FXDAT3: -1
15426  055772  052525                  FXDAT4: 052525
15427  055774  031463                  FXDAT5: 031463
15428  055776  007417                  FXDAT6: 007417
15429  056000  000477                  FXDAT7: 000477
15430
15431
15432  056002                          FDONE:
  (1)  056002  004767  046544                JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                 ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                 ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                 ;THE USER TYPED CONTROL G?).
15433
15434
15440                                   ;*************************************************************************
  (2)                                   ;TEST 440       FSRC MODE 0 TEST
  (3)                                   ;*************************************************************************
  (2)  056006                           TS440:
15441  056006                           I1:
15442  056006  170011                         SETD                    ;SET FD.
15443  056010  012700  056274                 MOV     #IDATI0,R0
15444  056014  012701  056244                 MOV     #IPAT10,R1
15445  056020  012702  000004                 MOV     #4,R2
15446  056024  012120                  I2:    MOV     (R1)+,(R0)+     ;SET UP THE INPUT DATA BUFFER.
15447  056026  077202                         SOB     R2,I2
15448
15449  056030  012700  056274                 MOV     #IDATI0,R0      ;LOAD AC1
15450  056034  172510                         LDD     (R0),AC1
15451
15452  056036  012700  056254                 MOV     #IPAT20,R0      ;LOAD AC0
15453  056042  172410                         LDD     (R0),AC0
15454
15455  056044  012701  000001                 MOV     #1,R1           ;IN CASE THE FSRC FLOWS FAIL
15456  056050  172401                  I3:    LDD     AC1,AC0         ;TEST INSTRUCTION.
15457  056052  000240                  I4:    NOP
15458  056054  000240                  I5:    NOP
15459
15460  056056  012700  056264                 MOV     #IDAT00,R0
15461  056062  174010                         STD     AC0,(R0)        ;GET AC0, THE RESULTS.
15462
15463  056064  012700  056264                 MOV     #IDAT00,R0      ;SEE IF DATA IS CORRECT.
15464  056070  012701  056274                 MOV     #IDATI0,R1
15465  056074  012702  000004                 MOV     #4,R2
15466  056100  022021                  I6:    CMP     (R0)+,(R1)+
15467  056102  001401                         BEQ     I105
```

M3



```
  (2)  056104  104000                          EMT                     ;
15468  056106  077204                  I105:   SOB     R2,I6
15469
15470                                  ;NOW TEST THE LOAD INSTRUCTION WITH FSRC MODE ZERO AND FD CLEAR.
15471
15472  056110  012700  056244          I12:    MOV     #IPAT10,R0
15473  056114  012701  056274                  MOV     #IDATI0,R1
15474  056120  012702  000004                  MOV     #4,R2
15475  056124  012021                  I13:    MOV     (R0)+,(R1)+
15476  056126  077202                          SOB     R2,I13
15477
15478  056130  012700  056274                  MOV     #IDATI0,R0              ;SET UP AC1
15479  056134  172510                          LDD     (R0),AC1
15480
15481  056136  012700  056254                  MOV     #IPAT20,R0              ;SET UP AC0
15482  056142  172410                          LDD     (R0),AC0
15483
15484  056144  012701  000001                  MOV     #1,R1
15485  056150  170001                          SETF                            ;CLEAR FD.
15486
15487  056152  172401                  I14:    LDF     AC1,AC0                 ;TEST INSTRUCTION.
15488  056154  000240                  I15:    NOP
15489  056156  000240                  I16:    NOP
15490
15491  056160  170200                          STFPS   R0                      ;SEE IF FPS IS STILL CLEAR.
15492  056162  022700  000004                  CMP     #4,R0
15493  056166  001401                          BEQ     I17
  (2)  056170  104000                          EMT                     ;
15494  056172                          I17:                                    ;RESET TO DOUBLE MODE.
15495  056172  170011                          SETD
15496
15497  056174  012700  056264                  MOV     #IDATO0,R0
15498  056200  174010                          STD     AC0,(R0)                ;GET AC0
15499
15500  056202  012737  177777  056300          MOV     #-1,@#IDATI2
15501  056210  012737  177777  056302          MOV     #-1,@#IDATI3
15502  056216  012700  056264                  MOV     #IDATO0,R0
15503  056222  012701  056274                  MOV     #IDATI0,R1
15504  056226  012702  000004                  MOV     #4,R2
15505  056232  022021                  I20:    CMP     (R0)+,(R1)+             ;SEE IF AC0 WAS CORRECT.
15506  056234  001401                          BEQ     I23
  (2)  056236  104000                          EMT                     ;
15507  056240  077204                  I23:    SOB     R2,I20
15508  056242  000420                          BR      IDONE                   ;NO ERRORS.
15509
15510  056244  000000                  IPAT10: 0
15511  056246  170360                  IPAT11: 170360
15512  056250  016161                  IPAT12: 016161
15513  056252  052525                  IPAT13: 052525
15514
15515  056254  177777                  IPAT20: -1
15516  056256  177777                  IPAT21: -1
15517  056260  177777                  IPAT22: -1
15518  056262  177777                  IPAT23: -1
15519
15520  056264  000000                  IDATO0: 0
```

```
15521  056266  000000                  IDATO1: 0
15522  056270  000000                  IDATO2: 0
15523  056272  000000                  IDATO3: 0
15524
15525  056274  000000                  IDATI0: 0
15526  056276  000000                  IDATI1: 0
15527  056300  000000                  IDATI2: 0
15528  056302  000000                  IDATI3: 0
15529
15530  056304                          IDONE:
  (1)  056304  004767  046242                JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                 ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                 ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                 ;THE USER TYPED CONTROL G?).
15531
15537
15538                                  ;*************************************************************************************
  (2)                                  ;TEST 441       FDST MODE 0 TEST
  (3)                                  ;*************************************************************************************
  (2)  056310                          TS441:
15539  056310  170011                          SETD                            ;SET FD
15540  056312  012700  056550                  MOV     #TPAT10,R0
15541  056316  012701  056600                  MOV     #TDATI0,R1
```

```
 15543  056322  012702  000004                    MOV     #4,R2
```

```
15545  056326  012021                  T2:     MOV     (R0)+,(R1)+             ;SET UP THE INPUT DATA BUFFER.
15546  056330  077202                          SOB     R2,T2
15547
```

```
15549  056332  012700  056600          MOV     #TDATIO,R0              ;LOAD AC0
15550  056336  172410                  LDD     (R0),AC0
15551
15552  056340  012700  056560          MOV     #TPAT20,R0              ;LOAD AC1
15553  056344  172510                  LDD     (R0),AC1
15554
15555  056346  012701  000001          MOV     #1,R1                   ;IF THE (BUT FDST) FORK FAILS
15556  056352  174001          T3:     STD     AC0,AC1
15557  056354  000240          T4:     NOP
15558  056356  000240          T5:     NOP
15559
15560  056360  012700  056570          MOV     #TDATOO,R0
15561  056364  174110                  STD     AC1,(R0)                ;GET THE DATA.
15562
15563  056366  012703  056570          MOV     #TDATOO,R3              ;SEE IF THE DATA IS CORRECT.
15564  056372  012704  056600          MOV     #TDATIO,R4
15565  056376  012705  000004          MOV     #4,R5
15566  056402  022324          T6:     CMP     (R3)+,(R4)+
15567  056404  001401                  BEQ     T105
  (2)  056406  104000                  EMT                             ;
15568  056410  077504          T105:   SOB     R5,T6
15569
15570                           ;NOW TEST THE STF AC0,AC1 INSTRUCTION.
15571
15572  056412  012700  056550  T12:    MOV     #TPAT10,R0              ;SET UP THE INPUT DATA BUFFER.
15573  056416  012701  056600          MOV     #TDATIO,R1
15574  056422  012702  000004          MOV     #4,R2
15575  056426  012021          T13:    MOV     (R0)+,(R1)+
15576  056430  077202                  SOB     R2,T13
15577
15578  056432  012700  056600          MOV     #TDATIO,R0              ;SET UP AC0
15579  056436  172410                  LDD     (R0),AC0
15580
15581  056440  012700  056560          MOV     #TPAT20,R0              ;SET UP AC1
15582  056444  172510                  LDD     (R0),AC1
15583
15584  056446  012701  000001          MOV     #1,R1
15585  056452  170001                  SETF                            ;CLEAR FD
15586  056454  174001          T14:    STF     AC0,AC1
15587  056456  000240          T15:    NOP
15588  056460  000240          T16:    NOP
15589
15590  056462  005000                  CLR     R0
15591  056464  170200                  STFPS   R0                      ;SEE IF FPS IS CLEAR.
15592  056466  022700  000010          CMP     #10,R0
15593  056472  001401                  BEQ     T17
  (2)  056474  104000                  EMT                             ;
15594  056476          T17:
15595  056476  170011                  SETD                            ;SET FD.
15596
15597  056500  012700  056570          MOV     #TDATOO,R0
15598  056504  174110                  STD     AC1,(R0)                ;PICK UP AC1.
15599
15600  056506  012737  177777  056604  MOV     #-1,@#TDATI2
15601  056514  012737  177777  056606  MOV     #-1,@#TDATI3
15602  056522  012703  056570          MOV     #TDATOO,R3
```

```
15603  056526  012704  056600                MOV     @TDATI0,R4
15604  056532  012705  000004                MOV     #4,R5
15605  056536  022324              T20:      CMP     (R3)+,(R4)+          ;WAS THE DATA TRANSFERRED CORRECTLY?
15606  056540  001401                        BEQ     T23
  (2)  056542  104000                        EMT                          ;
15607  056544  077504              T23:      SOB     R5,T20
15608  056546  000420                        BR      TDONE
15609
15610
15611  056550  000000              TPAT10: 0
15612  056552  170360              TPAT11: 170360
15613  056554  016161              TPAT12: 016161
15614  056556  052525              TPAT13: 052525
15615
15616  056560  177777              TPAT20: -1
15617  056562  177777              TPAT21: -1
15618  056564  177777              TPAT22: -1
15619  056566  177777              TPAT23: -1
15620
15621  056570  000000              TDAT00: 0
15622  056572  000000              TDAT01: 0
15623  056574  000000              TDAT02: 0
15624  056576  000000              TDAT03: 0
15625
15626  056600  000000              TDATI0: 0
15627  056602  000000              TDATI1: 0
15628  056604  000000              TDATI2: 0
15629  056606  000000              TDATI3: 0
15630
15631  056610                      TDONE:
  (1)  056610  004767  045736              JSR     PC,..RSET            ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                   ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                   ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                   ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                   ;THE USER TYPED CONTROL G?).
15632
15633
15644
15645                         ;*************************************************************************************
  (2)                         ;TEST 442       ACCUMULATORS DATA PATTERNS TEST
  (3)                         ;*************************************************************************************
  (2)  056614                 TS442:
15646  056614  170011                      SETD                                 ;SET FD.
15647                         ;TEST ACCUMULATOR 0 WITH FLOATING ONE
  (1)  056616  012700  060350              MOV     #GPAT00,R0
  (1)  056622  012701  060410              MOV     #GDAT00,R1
  (1)  056626  004737  060222              JSR     PC,@#GSETUP                  ;LOAD TEST PATTERN.
  (1)  056632  012703  000102              MOV     #102,R3
  (1)  056636                 G1:
  (1)  056636  172410                      LDD     (R0),AC0
  (1)  056640  174000                      STD     AC0,AC0
  (1)  056642  172400                      LDD     AC0,AC0              ;STORE THE TEST PATTERN.
  (1)  056644  174011                      STD     AC0,(R1)
  (1)  056646  004737  060320              JSR     PC,@#GCMP                    ;COMPARE THE DATA READ WITH
  (1)                                                                   ;THAT WHICH WAS WRITTEN.
  (1)  056652  005737  060344              TST     @#GFLAG1
```

```
   (1)  056656  001004                        BNE     G2
   (1)  056660  005137  060344                COM     @@GFLAG1
   (1)  056664  000261                        SEC
   (1)  056666  000401                        BR      G3
   (1)  056670  000241                G2:     CLC
   (1)  056672  006160  000006        G3:     ROL     6(R0)                   ;GENER .E THE NEXT TEST PATTERN.
   (1)  056676  006160  000004                ROL     4(R0)
   (1)  056702  006160  000002                ROL     2(R0)
   (1)  056706  006110                        ROL     (R0)
   (1)  056710  004737  060300                JSR     PC,@@GRESET             ;RESET DEFAULT PATTERN IN OUTPUT
   (1)                                                                         ;BUFFER.
   (1)  056714  077330                        SOB     R3,G1
   (1)
15648                                 ;TEST ACCUMULATOR 0 WITH FLOATING ZERO
   (1)  056716  012700  060360                MOV     #GPAT10,R0
   (1)  056722  012701  060410                MOV     #GDAT00,R1
   (1)  056726  004737  060222                JSR     PC,@@GSETUP             ;LOAD TEST PATTERN.
   (1)  056732  012703  000102                MOV     #102,R3
   (1)  056736                        G4:
   (1)  056736  172410                        LDD     (R0),AC0
   (1)  056740  174000                        STD     AC0,AC0
   (1)  056742  172400                        LDD     AC0,AC0         ;STORE THE TEST PATTERN.
   (1)  056744  174011                        STD     AC0,(R1)
   (1)  056746  004737  060320                JSR     PC,@@GCMP               ;COMPARE THE DATA READ WITH
   (1)                                                                         ;THAT WHICH WAS WRITTEN.
   (1)  056752  005737  060344                TST     @@GFLAG1
   (1)  056756  001004                        BNE     G5
   (1)  056760  005137  060344                COM     @@GFLAG1
   (1)  056764  000241                        CLC
   (1)  056766  000401                        BR      G6
   (1)  056770  000261                G5:     SEC
   (1)  056772  006160  000006        G6:     ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
   (1)  056776  006160  000004                ROL     4(R0)
   (1)  057002  006160  000002                ROL     2(R0)
   (1)  057006  006110                        ROL     (R0)
   (1)  057010  004737  060300                JSR     PC,@@GRESET             ;RESET DEFAULT PATTERN IN OUTPUT
   (1)                                                                         ;BUFFER.
   (1)  057014  077330                        SOB     R3,G4
   (1)
15649                                 ;TEST ACCUMULATOR 1 WITH FLOATING ONE
   (1)  057016  012700  060350                MOV     #GPAT00,R0
   (1)  057022  012701  060410                MOV     #GDAT00,R1
   (1)  057026  004737  060222                JSR     PC,@@GSETUP             ;LOAD TEST PATTERN.
   (1)  057032  012703  000102                MOV     #102,R3
   (1)  057036                        G7:
   (1)  057036  172410                        LDD     (R0),AC0
   (1)  057040  174001                        STD     AC0,AC1
   (1)  057042  172401                        LDD     AC1,AC0         ;STORE THE TEST PATTERN.
   (1)  057044  174011                        STD     AC0,(R1)
   (1)  057046  004737  060320                JSR     PC,@@GCMP               ;COMPARE THE DATA READ WITH
   (1)                                                                         ;THAT WHICH WAS WRITTEN.
   (1)  057052  005737  060344                TST     @@GFLAG1
   (1)  057056  001004                        BNE     G10
   (1)  057060  005137  060344                COM     @@GFLAG1
   (1)  057064  000261                        SEC
   (1)  057066  000401                        BR      G11
```

```
   (1)  057070  000241                 G10:    CLC
   (1)  057072  006160  000006         G11:    ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
   (1)  057076  006160  000004                 ROL     4(R0)
   (1)  057102  006160  000002                 ROL     2(R0)
   (1)  057106  006110                         ROL     (R0)
   (1)  057110  004737  060300                 JSR     PC,@#GRESET             ;RESET DEFAULT PATTERN IN OUTPUT
   (1)                                                                         ;BUFFER.
   (1)  057114  077330                         SOB     R3,G7
   (1)
15650                                  ;TEST ACCUMULATOR 1 WITH FLOATING ZERO
   (1)  057116  012700  060360                 MOV     #GPAT10,R0
   (1)  057122  012701  060410                 MOV     #GDAT00,R1
   (1)  057126  004737  060222                 JSR     PC,@#GSETUP             ;LOAD TEST PATTERN.
   (1)  057132  012703  000102                 MOV     #102,R3
   (1)  057136                            G12:
   (1)  057136  172410                         LDD     (R0),AC0
   (1)  057140  174001                         STD     AC0,AC1
   (1)  057142  172401                         LDD     AC1,AC0         ;STORE THE TEST PATTERN.
   (1)  057144  174011                         STD     AC0,(R1)
   (1)  057146  004737  060320                 JSR     PC,@#GCMP               ;COMPARE THE DATA READ WITH
   (1)                                                                         ;THAT WHICH WAS WRITTEN.
   (1)  057152  005737  060344                 TST     @#GFLAG1
   (1)  057156  001004                         BNE     G13
   (1)  057160  005137  060344                 COM     @#GFLAG1
   (1)  057164  000241                         CLC
   (1)  057166  000401                         BR      G14
   (1)  057170  000261                    G13: SEC
   (1)  057172  006160  000006         G14:    ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
   (1)  057176  006160  000004                 ROL     4(R0)
   (1)  057202  006160  000002                 ROL     2(R0)
   (1)  057206  006110                         ROL     (R0)
   (1)  057210  004737  060300                 JSR     PC,@#GRESET             ;RESET DEFAULT PATTERN IN OUTPUT
   (1)                                                                         ;BUFFER.
   (1)  057214  077330                         SOB     R3,G12
   (1)
15651                                  ;TEST ACCUMULATOR 2 WITH FLOATING ONE
   (1)  057216  012700  060350                 MOV     #GPAT00,R0
   (1)  057222  012701  060410                 MOV     #GDAT00,R1
   (1)  057226  004737  060222                 JSR     PC,@#GSETUP             ;LOAD TEST PATTERN.
   (1)  057232  012703  000102                 MOV     #102,R3
   (1)  057236                            G15:
   (1)  057236  172410                         LDD     (R0),AC0
   (1)  057240  174002                         STD     AC0,AC2
   (1)  057242  172402                         LDD     AC2,AC0         ;STORE THE TEST PATTERN.
   (1)  057244  174011                         STD     AC0,(R1)
   (1)  057246  004737  060320                 JSR     PC,@#GCMP               ;COMPARE THE DATA READ WITH
   (1)                                                                         ;THAT WHICH WAS WRITTEN.
   (1)  057252  005737  060344                 TST     @#GFLAG1
   (1)  057256  001004                         BNE     G16
   (1)  057260  005137  060344                 COM     @#GFLAG1
   (1)  057264  000261                         SEC
   (1)  057266  000401                         BR      G17
   (1)  057270  000241                    G16: CLC
   (1)  057272  006160  000006         G17:    ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
   (1)  057276  006160  000004                 ROL     4(R0)
   (1)  057302  006160  000002                 ROL     2(R0)
```

```
  (1)  057306  006110                    ROL     (R0)
  (1)  057310  004737  060300            JSR     PC,@#GRESET             ;RESET DEFAULT PATTERN IN OUTPUT
  (1)                                                                    ;BUFFER.
  (1)  057314  077330                    SOB     R3,G15
  (1)
15652                                ;TEST ACCUMULATOR 2 WITH FLOATING ZERO
  (1)  057316  012700  060360            MOV     #GPAT10,R0
  (1)  057322  012701  060410            MOV     #GDAT00,R1
  (1)  057326  004737  060222            JSR     PC,@#GSETUP             ;LOAD TEST PATTERN.
  (1)  057332  012703  000102            MOV     #102,R3
  (1)  057336                        G20:
  (1)  057336  172410                    LDD     (R0),AC0
  (1)  057340  174002                    STD     AC0,AC2
  (1)  057342  172402                    LDD     AC2,AC0         ;STORE THE TEST PATTERN.
  (1)  057344  174011                    STD     AC0,(R1)
  (1)  057346  004737  060320            JSR     PC,@#GCMP               ;COMPARE THE DATA READ WITH
  (1)                                                                    ;THAT WHICH WAS WRITTEN.
  (1)  057352  005737  060344            TST     @#GFLAG1
  (1)  057356  001004                    BNE     G21
  (1)  057360  005137  060344            COM     @#GFLAG1
  (1)  057364  000241                    CLC
  (1)  057366  000401                    BR      G22
  (1)  057370  000261                G21: SEC
  (1)  057372  006160  000006        G22: ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
  (1)  057376  006160  000004            ROL     4(R0)
  (1)  057402  006160  000002            ROL     2(R0)
  (1)  057406  006110                    ROL     (R0)
  (1)  057410  004737  060300            JSR     PC,@#GRESET             ;RESET DEFAULT PATTERN IN OUTPUT
  (1)                                                                    ;BUFFER.
  (1)  057414  077330                    SOB     R3,G20
  (1)
15653                                ;TEST ACCUMULATOR 3 WITH FLOATING ONE
  (1)  057416  012700  060350            MOV     #GPAT00,R0
  (1)  057422  012701  060410            MOV     #GDAT00,R1
  (1)  057426  004737  060222            JSR     PC,@#GSETUP             ;LOAD TEST PATTERN.
  (1)  057432  012703  000102            MOV     #102,R3
  (1)  057436                        G23:
  (1)  057436  172410                    LDD     (R0),AC0
  (1)  057440  174003                    STD     AC0,AC3
  (1)  057442  172403                    LDD     AC3,AC0         ;STORE THE TEST PATTERN.
  (1)  057444  174011                    STD     AC0,(R1)
  (1)  057446  004737  060320            JSR     PC,@#GCMP               ;COMPARE THE DATA READ WITH
  (1)                                                                    ;THAT WHICH WAS WRITTEN.
  (1)  057452  005737  060344            TST     @#GFLAG1
  (1)  057456  001004                    BNE     G24
  (1)  057460  005137  060344            COM     @#GFLAG1
  (1)  057464  000261                    SEC
  (1)  057466  000401                    BR      G25
  (1)  057470  000241                G24: CLC
  (1)  057472  006160  000006        G25: ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
  (1)  057476  006160  000004            ROL     4(R0)
  (1)  057502  006160  000002            ROL     2(R0)
  (1)  057506  006110                    ROL     (R0)
  (1)  057510  004737  060300            JSR     PC,@#GRESET             ;RESET DEFAULT PATTERN IN OUTPUT
  (1)                                                                    ;BUFFER.
  (1)  057514  077330                    SOB     R3,G23
```

```
  (1)
15654                                   ;TEST ACCUMULATOR 3 WITH FLOATING ZERO
  (1)  057516  012700  060360                   MOV     #GPAT10,R0
  (1)  057522  012701  060410                   MOV     #GDAT00,R1
  (1)  057526  004737  060222                   JSR     PC,@#GSETUP             ;LOAD TEST PATTERN.
  (1)  057532  012703  000102                   MOV     #102,R3
  (1)  057536                           G26:
  (1)  057536  172410                           LDD     (R0),AC0
  (1)  057540  174003                           STD     AC0,AC3
  (1)  057542  172403                           LDD     AC3,AC0         ;STORE THE TEST PATTERN.
  (1)  057544  174011                           STD     AC0,(R1)
  (1)  057546  004737  060320                   JSR     PC,@#GCMP               ;COMPARE THE DATA READ WITH
  (1)                                                                   ;THAT WHICH WAS WRITTEN.
  (1)  057552  005737  060344                   TST     @#GFLAG1
  (1)  057556  001004                           BNE     G27
  (1)  057560  005137  060344                   COM     @#GFLAG1
  (1)  057564  000241                           CLC
  (1)  057566  000401                           BR      G30
  (1)  057570  000261                   G27:    SEC
  (1)  057572  006160  000006           G30:    ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
  (1)  057576  006160  000004                   ROL     4(R0)
  (1)  057602  006160  000002                   ROL     2(R0)
  (1)  057606  006110                           ROL     (R0)
  (1)  057610  004737  060300                   JSR     PC,@#GRESET             ;RESET DEFAULT PATTERN IN OUTPUT
  (1)                                                                   ;BUFFER.
  (1)  057614  077330                           SOB     R3,G26
  (1)
15655                                   ;TEST ACCUMULATOR 4 WITH FLOATING ONE
  (1)  057616  012700  060350                   MOV     #GPAT00,R0
  (1)  057622  012701  060410                   MOV     #GDAT00,R1
  (1)  057626  004737  060222                   JSR     PC,@#GSETUP             ;LOAD TEST PATTERN.
  (1)  057632  012703  000102                   MOV     #102,R3
  (1)  057636                           G31:
  (1)  057636  172410                           LDD     (R0),AC0
  (1)  057640  174004                           STD     AC0,AC4
  (1)  057642  172404                           LDD     AC4,AC0         ;STORE THE TEST PATTERN.
  (1)  057644  174011                           STD     AC0,(R1)
  (1)  057646  004737  060320                   JSR     PC,@#GCMP               ;COMPARE THE DATA READ WITH
  (1)                                                                   ;THAT WHICH WAS WRITTEN.
  (1)  057652  005737  060344                   TST     @#GFLAG1
  (1)  057656  001004                           BNE     G32
  (1)  057660  005137  060344                   COM     @#GFLAG1
  (1)  057664  000261                           SEC
  (1)  057666  000401                           BR      G33
  (1)  057670  000241                   G32:    CLC
  (1)  057672  006160  000006           G33:    ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
  (1)  057676  006160  000004                   ROL     4(R0)
  (1)  057702  006160  000002                   ROL     2(R0)
  (1)  057706  006110                           ROL     (R0)
  (1)  057710  004737  060300                   JSR     PC,@#GRESET             ;RESET DEFAULT PATTERN IN OUTPUT
  (1)                                                                   ;BUFFER.
  (1)  057714  077330                           SOB     R3,G31
  (1)
15656                                   ;TEST ACCUMULATOR 4 WITH FLOATING ZERO
  (1)  057716  012700  060360                   MOV     #GPAT10,R0
  (1)  057722  012701  060410                   MOV     #GDAT00,R1
```

```
  (1)  057726  004737  060222                 JSR     PC,@#GSETUP             ;LOAD TEST PATTERN.
  (1)  057732  012703  000102                 MOV     #102,R3
  (1)  057736                         G34:
  (1)  057736  172410                         LDD     (R0),AC0
  (1)  057740  174004                         STD     AC0,AC4
  (1)  057742  172404                         LDD     AC4,AC0         ;STORE THE TEST PATTERN.
  (1)  057744  174011                         STD     AC0,(R1)
  (1)  057746  004737  060320                 JSR     PC,@#GCMP               ;COMPARE THE DATA READ WITH
  (1)                                                                         ;THAT WHICH WAS WRITTEN.
  (1)  057752  005737  060344                 TST     @#GFLAG1
  (1)  057756  001004                         BNE     G35
  (1)  057760  005137  060344                 COM     @#GFLAG1
  (1)  057764  000241                         CLC
  (1)  057766  000401                         BR      G36
  (1)  057770  000261                 G35:    SEC
  (1)  057772  006160  000006        G36:    ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
  (1)  057776  006160  000004                 ROL     4(R0)
  (1)  060002  006160  000002                 ROL     2(R0)
  (1)  060006  006110                         ROL     (R0)
  (1)  060010  004737  060300                 JSR     PC,@#GRESET             ;RESET DEFAULT PATTERN IN OUTPUT
  (1)                                                                         ;BUFFER.
  (1)  060014  077330                         SOB     R3,G34
  (1)
15657                                 ;TEST ACCUMULATOR 5 WITH FLOATING ONE
  (1)  060016  012700  060350                   MOV     #GPAT00,R0
  (1)  060022  012701  060410                   MOV     #GDAT00,R1
  (1)  060026  004737  060222                   JSR     PC,@#GSETUP             ;LOAD TEST PATTERN.
  (1)  060032  012703  000102                   MOV     #102,R3
  (1)  060036                           G37:
  (1)  060036  172410                           LDD     (R0),AC0
  (1)  060040  174005                           STD     AC0,AC5
  (1)  060042  172405                           LDD     AC5,AC0         ;STORE THE TEST PATTERN.
  (1)  060044  174011                           STD     AC0,(R1)
  (1)  060046  004737  060320                   JSR     PC,@#GCMP               ;COMPARE THE DATA READ WITH
  (1)                                                                           ;THAT WHICH WAS WRITTEN.
  (1)  060052  005737  060344                   TST     @#GFLAG1
  (1)  060056  001004                           BNE     G40
  (1)  060060  005137  060344                   COM     @#GFLAG1
  (1)  060064  000261                           SEC
  (1)  060066  000401                           BR      G41
  (1)  060070  000241                   G40:    CLC
  (1)  060072  006160  000006          G41:    ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
  (1)  060076  006160  000004                   ROL     4(R0)
  (1)  060102  006160  000002                   ROL     2(R0)
  (1)  060106  006110                           ROL     (R0)
  (1)  060110  004737  060300                   JSR     PC,@#GRESET             ;RESET DEFAULT PATTERN IN OUTPUT
  (1)                                                                           ;BUFFER.
  (1)  060114  077330                           SOB     R3,G37
  (1)
15658                                 ;TEST ACCUMULATOR 5 WITH FLOATING ZERO
  (1)  060116  012700  060360                   MOV     #GPAT10,R0
  (1)  060122  012701  060410                   MOV     #GDAT00,R1
  (1)  060126  004737  060222                   JSR     PC,@#GSETUP             ;LOAD TEST PATTERN.
  (1)  060132  012703  000102                   MOV     #102,R3
  (1)  060136                           G42:
  (1)  060136  172410                           LDD     (R0),AC0
```

```
  (1)  060140  174005                        STD     AC0,AC5
  (1)  060142  172405                        LDD     AC5,AC0         ;STORE THE TEST PATTERN.
  (1)  060144  174011                        STD     AC0,(R1)
  (1)  060146  004737  060320                JSR     PC,@#GCMP               ;COMPARE THE DATA READ WITH
  (1)                                                                        ;THAT WHICH WAS WRITTEN.
  (1)
  (1)  060152  005737  060344                TST     @#GFLAG1
  (1)  060156  001004                        BNE     G43
  (1)  060160  005137  060344                COM     @#GFLAG1
  (1)  060164  000241                        CLC
  (1)  060166  000401                        BR      G44
  (1)  060170  000261                G43:    SEC
  (1)  060172  006160  000006       G44:    ROL     6(R0)                   ;GENERATE THE NEXT TEST PATTERN.
  (1)  060176  006160  000004               ROL     4(R0)
  (1)  060202  006160  000002               ROL     2(R0)
  (1)  060206  006110                        ROL     (R0)
  (1)  060210  004737  060300                JSR     PC,@#GRESET             ;RESET DEFAULT PATTERN IN OUTPUT
  (1)                                                                        ;BUFFER.
  (1)  060214  077330                        SOB     R3,G42
  (1)
15659
15660  060216  000137  060420                JMP     @#GDONE
15661
15662                                ;USE THIS ROUTINE TO INITIALIZE ALL THE DATA BUFFERS.
15663  060222  012705  060344       GSETUP: MOV     #GFLAG1,R5
15664  060226  012704  000026               MOV     #26,R4
15665  060232  005025                1$:     CLR     (R5)+
15666  060234  077402                        SOB     R4,1$
15667
15668  060236  012705  060360               MOV     #GPAT10,R5
15669  060242  012704  000010               MOV     #10,R4
15670  060246  005125                2$:     COM     (R5)+
15671  060250  077402                        SOB     R4,2$
15672
15673  060252  020067  000072       GS1:    CMP     R0,GPAT00
15674  060256  001401                        BEQ     3$
15675  060260  000207                        RTS     PC
15676
15677  060262  012705  060410       3$:     MOV     #GDAT00,R5
15678  060266  012704  000004               MOV     #4,R4
15679  060272  005125                4$:     COM     (R5)+
15680  060274  077402                        SOB     R4,4$
15681  060276  000207                        RTS     PC
15682
15683  060300  012705  060410       GRESET: MOV     #GDAT00,R5
15684  060304  012704  000004               MOV     #4,R4
15685  060310  005025                1$:     CLR     (R5)+
15686  060312  077402                        SOB     R4,1$
15687  060314  000137  060252               JMP     @#GS1
15688
15689                                ;SEE IF THE DATA WRITTEN MATCHES THE DATA READ.
15690  060320  012705  060410       GCMP:   MOV     #GDAT00,R5
15691  060324  012704  000004               MOV     #4,R4
15692  060330  010002                        MOV     R0,R2
15693  060332  022225                1$:     CMP     (R2)+,(R5)+
15694  060334  001401                        BEQ     2$
  (2)  060336  104000                        EMT                             ;
```

```
15695  060340  077404                  2$:     SOB     R4,1$
15696  060342  000207                          RTS     PC
15697
15698
15699
15700
15701  060344  000000                  GFLAG1: 0
15702  060346  000000                  GFLAG2: 0
15703
15704  060350  000000                  GPAT00: 0
15705  060352  000000                  GPAT01: 0
15706  060354  000000                  GPAT02: 0
15707  060356  000000                  GPAT03: 0
15708
15709  060360  177777                  GPAT10: -1
15710  060362  177777                  GPAT11: -1
15711  060364  177777                  GPAT12: -1
15712  060366  177777                  GPAT13: -1
15713
15714  060370  177777                  GAND0:  -1
15715  060372  177777                  GAND1:  -1
15716  060374  177777                  GAND2:  -1
15717  060376  177777                  GAND3:  -1
15718
15719  060400  000000                  GOR0:   0
15720  060402  000000                  GOR1:   0
15721  060404  000000                  GOR2:   0
15722  060406  000000                  GOR3:   0
15723
15724  060410  000000                  GDAT00: 0
15725  060412  000000                  GDAT01: 0
15726  060414  000000                  GDAT02: 0
15727  060416  000000                  GDAT03: 0
15728
15729
15730  060420                          GDONE:
  (1)  060420  004767  044126                  JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
15731
15732
15739                                   ;*************************************************************************
  (2)                                   ;TEST 443       FPP ACCUMULATORS DUAL ADDRESS TEST
  (3)                                   ;*************************************************************************
  (2)  060424                           TS443:
15740  060424  005037  061114          H1:     CLR     @#HFLAG
15741  060430  012700  061116                  MOV     #HA1W,R0                ;INITIALIZE THE LOAD BUFFER DATA.
15742  060434  012701  061236                  MOV     #HDAT1,R1
15743  060440  012703  000024                  MOV     #24,R3
15744  060444  012120                  H2:     MOV     (R1)+,(R0)+
15745  060446  077302                          SOB     R3,H2
15746
15747  060450  004767  000420                  JSR     PC,HCLR         ;CLEAR THE OUTPUT DATA BUFFER.
15748
```

```
15749  060454  170011                  H3:     SETD
15750                                  ;LOAD ACCUMULATOR 1
  (1)  060456  012700  061116                  MOV     #HA1W,R0
  (1)  060462  172410                          LDD     (R0),AC0
  (1)  060464  174001                          STD     AC0,AC1
15751                                  ;LOAD ACCUMULATOR 2
  (1)  060466  012700  061126                  MOV     #HA2W,R0
  (1)  060472  172410                          LDD     (R0),AC0
  (1)  060474  174002                          STD     AC0,AC2
15752                                  ;LOAD ACCUMULATOR 3
  (1)  060476  012700  061136                  MOV     #HA3W,R0
  (1)  060502  172410                          LDD     (R0),AC0
  (1)  060504  174003                          STD     AC0,AC3
15753                                  ;LOAD ACCUMULATOR 4
  (1)  060506  012700  061146                  MOV     #HA4W,R0
  (1)  060512  172410                          LDD     (R0),AC0
  (1)  060514  174004                          STD     AC0,AC4
15754                                  ;LOAD ACCUMULATOR 5
  (1)  060516  012700  061156                  MOV     #HA5W,R0
  (1)  060522  172410                          LDD     (R0),AC0
  (1)  060524  174005                          STD     AC0,AC5
15755
15756  060526  004737  060762          H4:     JSR     PC,@#HSTD               ;GO READ ALL ACCUMULATORS BACK.
15757
15758  060532  004737  061040                  JSR     PC,@#HCMP               ;SEE IF DATA IS CORRECT.
15759
15760                                  ;COMPLIMENT EACH WORD OF THE DATA STORED IN ACCUMULATOR 1,
  (1)                                  ;RELOAD THAT ACCUMULATOR, READ ALL THE ACCUMULATORS BACK AND CHECK
  (1)                                  ;THE DATA.
  (1)  060536  012700  061116                  MOV     #HA1W,R0
  (1)  060542  012702  000004                  MOV     #4,R2
  (1)  060546  010001                          MOV     R0,R1
  (1)  060550  005121          H5:     COM     (R1)+
  (1)  060552  172410                          LDD     (R0),AC0
  (1)  060554  174001                          STD     AC0,AC1
  (1)  060556  004737  060762                  JSR     PC,@#HSTD               ;READ ALL THE ACCUMULATORS BACK.
  (1)  060562  004737  061040                  JSR     PC,@#HCMP               ;CHECK THE DATA.
  (1)  060566  077210                          SOB     R2,H5
  (1)
15761                                  ;COMPLIMENT EACH WORD OF THE DATA STORED IN ACCUMULATOR 2,
  (1)                                  ;RELOAD THAT ACCUMULATOR, READ ALL THE ACCUMULATORS BACK AND CHECK
  (1)                                  ;THE DATA.
  (1)  060570  012700  061126                  MOV     #HA2W,R0
  (1)  060574  012702  000004                  MOV     #4,R2
  (1)  060600  010001                          MOV     R0,R1
  (1)  060602  005121          H6:     COM     (R1)+
  (1)  060604  172410                          LDD     (R0),AC0
  (1)  060606  174002                          STD     AC0,AC2
  (1)  060610  004737  060762                  JSR     PC,@#HSTD               ;READ ALL THE ACCUMULATORS BACK.
  (1)  060614  004737  061040                  JSR     PC,@#HCMP               ;CHECK THE DATA.
  (1)  060620  077210                          SOB     R2,H6
  (1)
15762                                  ;COMPLIMENT EACH WORD OF THE DATA STORED IN ACCUMULATOR 3,
  (1)                                  ;RELOAD THAT ACCUMULATOR, READ ALL THE ACCUMULATORS BACK AND CHECK
  (1)                                  ;THE DATA.
  (1)  060622  012700  061136                  MOV     #HA3W,R0
```

```
  (1)  060626  012702  000004                MOV     #4,R2
  (1)  060632  010001                        MOV     R0,R1
  (1)  060634  005121                H7:     COM     (R1)+
  (1)  060636  172410                        LDD     (R0),AC0
  (1)  060640  174003                        STD     AC0,AC3
  (1)  060642  004737  060762                JSR     PC,@#HSTD               ;READ ALL THE ACCUMULATORS BACK.
  (1)  060646  004737  061040                JSR     PC,@#HCMP               ;CHECK THE DATA.
  (1)  060652  077210                        SOB     R2,H7
  (1)
15763                                ;COMPLIMENT EACH WORD OF THE DATA STORED IN ACCUMULATOR 4,
  (1)                                ;RELOAD THAT ACCUMULATOR, READ ALL THE ACCUMULATORS BACK AND CHECK
  (1)                                ;THE DATA.
  (1)  060654  012700  061146                MOV     #HA4W,R0
  (1)  060660  012702  000004                MOV     #4,R2
  (1)  060664  010001                        MOV     R0,R1
  (1)  060666  005121                H10:    COM     (R1)+
  (1)  060670  172410                        LDD     (R0),AC0
  (1)  060672  174004                        STD     AC0,AC4
  (1)  060674  004737  060762                JSR     PC,@#HSTD               ;READ ALL THE ACCUMULATORS BACK.
  (1)  060700  004737  061040                JSR     PC,@#HCMP               ;CHECK THE DATA.
  (1)  060704  077210                        SOB     R2,H10
  (1)
15764                                ;COMPLIMENT EACH WORD OF THE DATA STORED IN ACCUMULATOR 5,
  (1)                                ;RELOAD THAT ACCUMULATOR, READ ALL THE ACCUMULATORS BACK AND CHECK
  (1)                                ;THE DATA.
  (1)  060706  012700  061156                MOV     #HA5W,R0
  (1)  060712  012702  000004                MOV     #4,R2
  (1)  060716  010001                        MOV     R0,R1
  (1)  060720  005121                H11:    COM     (R1)+
  (1)  060722  172410                        LDD     (R0),AC0
  (1)  060724  174005                        STD     AC0,AC5
  (1)  060726  004737  060762                JSR     PC,@#HSTD               ;READ ALL THE ACCUMULATORS BACK.
  (1)  060732  004737  061040                JSR     PC,@#HCMP               ;CHECK THE DATA.
  (1)  060736  077210                        SOB     R2,H11
  (1)
15765
15766  060740  005737  061114                TST     @#HFLAG
15767  060744  001402                        BEQ     H12
15768  060746  000137  061306                JMP     @#HDONE
15769
15770  060752  005137  061114          H12:  COM     @#HFLAG
15771  060756  000137  060454                JMP     @#H3
15772
15773                                ;STORE ALL ACCUMULATORS IN THE OUTPUT BUFFERS.
15774  060762  004737  061074          HSTD:  JSR     PC,@#HCLR               ;CLEAR ALL OUTPUT BUFFERS.
15775                                ;STORE ACCUMULATOR 1
  (1)  060766  012704  061166                MOV     #HA1R,R4
  (1)  060772  172401                        LDD     AC1,AC0
  (1)  060774  174014                        STD     AC0,(R4)
15776                                ;STORE ACCUMULATOR 2
  (1)  060776  012704  061176                MOV     #HA2R,R4
  (1)  061002  172402                        LDD     AC2,AC0
  (1)  061004  174014                        STD     AC0,(R4)
15777                                ;STORE ACCUMULATOR 3
  (1)  061006  012704  061206                MOV     #HA3R,R4
  (1)  061012  172403                        LDD     AC3,AC0
```

```
  (1)  061014  174014                        STD     AC0,(R4)
15778                                ;STORE ACCUMULATOR 4
  (1)  061016  012704  061216                MOV     @#A4R,R4
  (1)  061022  172404                        LDD     AC4,AC0
  (1)  061024  174014                        STD     AC0,(R4)
15779                                ;STORE ACCUMULATOR 5
  (1)  061026  012704  061226                MOV     @#A5R,R4
  (1)  061032  172405                        LDD     AC5,AC0
  (1)  061034  174014                        STD     AC0,(R4)
15780  061036  000207                        RTS     PC
15781
15782                                ;COMPARE DATA LOADED WITH DATA READ.
15783  061040  012637  061112        MCMP:   MOV     (SP)+,@#MADR            ;SAVE RETURN ADDRESS.
15784  061044  012703  061116                MOV     @#A1W,R3
15785  061050  012704  061166                MOV     @#A1R,R4
15786  061054  012705  000024                MOV     #24,R5
15787  061060  022324                MCMP1:  CMP     (R3)+,(R4)+
15788  061062  001401                        BEQ     MCMP2
  (2)  061064  104000                        EMT                             ;
15789  061066  077504                MCMP2:  SOB     R5,MCMP1
15790  061070  000177  000016                JMP     @#MADR
15791
15792                                ;CLEAR THE DATA OUTPUT BUFFER.
15793  061074  012704  061166        MCLR:   MOV     @#A1R,R4
15794  061100  012705  000024                MOV     #24,R5
15795  061104  005024                MCLR1:  CLR     (R4)+
15796  061106  077502                        SOB     R5,MCLR1
15797  061110  000207                        RTS     PC
15798
15799  061112  000000                MADR:   0
15800  061114  000000                MFLAG:  0
15801
15802  061116  000000  000000  000000  MA1W:   .WORD   0,0,0,0
       061124  000000
15803  061126  000000  000000  000000  MA2W:   .WORD   0,0,0,0
       061134  000000
15804  061136  000000  000000  000000  MA3W:   .WORD   0,0,0,0
       061144  000000
15805  061146  000000  000000  000000  MA4W:   .WORD   0,0,0,0
       061154  000000
15806  061156  000000  000000  000000  MA5W:   .WORD   0,0,0,0
       061164  000000
15807
15808  061166  000000  000000  000000  MA1R:   .WORD   0,0,0,0
       061174  000000
15809  061176  000000  000000  000000  MA2R:   .WORD   0,0,0,0
       061204  000000
15810  061206  000000  000000  000000  MA3R:   .WORD   0,0,0,0
       061214  000000
15811  061216  000000  000000  000000  MA4R:   .WORD   0,0,0,0
       061224  000000
15812  061226  000000  000000  000000  MA5R:   .WORD   0,0,0,0
       061234  000000
15813
15814  061236  073567  073567  073567  MDAT1:  .WORD   73567,73567,73567,73567
       061244  073567
```

```
15815  061246  063146  063146  063146  HDAT2:  .WORD   63146,63146,63146,63146
       061254  063146
15816  061256  010421  010421  010421  HDAT3:  .WORD   10421,10421,10421,10421
       061264  010421
15817  061266  031463  031463  031463  HDAT4:  .WORD   31463,31463,31463,31463
       061274  031463
15818  061276  042104  042104  042104  HDAT5:  .WORD   42104,42104,42104,42104
       061304  042104
15819
15820  061306                          HDONE:
  (1)  061306  004767  043240                  JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
15821
15822
15830                           ;*************************************************************************
  (2)                           ;TEST 444       FSRC MODE 0 WITH ILLEGAL ACCUMULATOR TEST
  (3)                           ;*************************************************************************
  (2)  061312                   TS444:
15831  061312  170011                          SETD                    ;SET FD
15832  061314  012700  061574                  MOV     #SPAT10,R0      ;LOAD AC0
15833  061320  172410                          LDD     (R0),AC0
15834
15835  061322  012737  061500  000244          MOV     #SERR0,@#FPVECT ;USE OF THE NON-EXISTENT AC-
15836                                                                   ;CUMULATOR SHOULD RESULT IN
15837                                                                   ;A TRAP TO 244.
15838  061330  012700  000001                  MOV     #1,R0           ;A FAILURE IN THE FSRC FLOWS
15839                                                                   ;WILL RESULT IN AN ODD ADDRESS
15840  061334  012737  061406  000004          MOV     #SERR1,@#ERRVECT ;TRAP TO 4.
15841  061342  005003                          CLR     R3
15842
15843  061344  172407                  SX2:    LDD     AC7,AC0
15844  061346  170000                  SX3:    CFCC
15845  061350  005203                          INC     R3
15846  061352  005203                  SX4:    INC     R3
15847
15848  061354  012701  061604                  MOV     #SDAT00,R1              ;NO TRAP OCCURRED!!
15849  061360  174011                          STD     AC0,(R1)                ;SEE IF AC0 WAS MODIFIED.
15850
15851  061362  012701  061604                  MOV     #SDAT00,R1
15852  061366  012702  061574                  MOV     #SPAT10,R2
15853  061372  012703  000004                  MOV     #4,R3
15854  061376  022122                  SX5:    CMP     (R1)+,(R2)+
15855  061400  001401                          BEQ     SX6
  (2)  061402  104000                          EMT                     ;
15856  061404  077304                  SX6:    SOB     R3,SX5
15857  061406                          SERR1:
  (2)  061406  104000                          EMT                     ;
15858
15859                                   ;NOW TEST AC6.
15860  061410  170011                  SX7:    SETD
15861  061412  012700  061574                  MOV     #SPAT10,R0              ;LOAD AC0
15862  061416  172410                          LDD     (R0),AC0
15863  061420  012737  061550  000244          MOV     #SERR4,@#FPVECT
```

```
15864  061426  012700  000001              MOV     #1,R0
15865  061432  005003                      CLR     R3
15866
15867  061434  172406              SX8:    LDD     AC6,AC0
15868  061436  170000              SX9:    CFCC
15869  061440  005203                      INC     R3
15870  061442  005203              SX10:   INC     R3
15871
15872  061444  012701  061604              MOV     #SDAT00,R1
15873  061450  174011                      STD     AC0,(R1)                ;NO TRAP! GET AC0.
15874
15875  061452  012701  061604              MOV     #SDAT00,R1              ;WAS AC0 MODIFIED.
15876  061456  012702  061574              MOV     #SPAT10,R2
15877  061462  012703  000004              MOV     #4,R3
15878  061466  022122              SX11:   CMP     (R1)+,(R2)+
15879  061470  001401                      BEQ     SX12
  (2)  061472  104000                      EMT                             ;
15880  061474  077304              SX12:   SOB     R3,SX11
15881  061476  104000                      EMT                             ;
15882
15883                               ;TRAPPED TO 244.
15884  061500  021627  061346      SERR0:  CMP     (SP),#SX3               ;PC OF TRAP CORRECT?
15885  061504  001401                      BEQ     1$
  (2)  061506  104000                      EMT                             ;
15886  061510  012737  061410  061570  1$: MOV     #SX7,@#SADR
15887  061516  022626              SERR10: CMP     (SP)+,(SP)+
15888  061520  005004                      CLR     R4
15889  061522  170204                      STFPS   R4                      ;IS FPS CORRECT?
15890  061524  022704  100200              CMP     #100200,R4
15891  061530  001326                      BNE     SERR1
15892
15893  061532  005004                      CLR     R4
15894  061534  170304                      STST    R4                      ;IS FEC CORRECT?
15895  061536  022704  000002              CMP     #2,R4
15896  061542  001321                      BNE     SERR1
15897  061544  000177  000020              JMP     @#SADR
15898
15899  061550  021627  061436      SERR4:  CMP     (SP),#SX9
15900  061554  001401                      BEQ     1$
  (2)  061556  104000                      EMT                             ;
15901  061560  012737  061614  061570  1$: MOV     #SDONE,@#SADR
15902  061566  000753                      BR      SERR10
15903
15904  061570  000000              SADR:   0
15905  061572  177777                      -1
15906  061574  010421              SPAT10: 10421
15907  061576  021042              SPAT11: 21042
15908  061600  031463              SPAT12: 31463
15909  061602  042104              SPAT13: 42104
15910
15911  061604  000000              SDAT00: 0
15912  061606  000000              SDAT01: 0
15913  061610  000000              SDAT02: 0
15914  061612  000000              SDAT03: 0
15915
15916  061614                      SDONE:
```

```
  (1)  061614  004767  042732                JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
15917
15924                                ;*************************************************************************************
  (2)                                ;TEST 445       FSRC MODE 2 TEST
  (3)                                ;*************************************************************************************
  (2)  061620                        TS445:
15925  061620                      J1:
15926  061620  170011                       SETD                    ;SET DOUBLE MODE
15927
15928  061622  012700  061750               MOV     #JDATO,R0
15929  061626  172410                       LDD     (R0),AC0        ;LOAD AC0=ALL 1
15930
15931  061630  012700  061730               MOV     #JDATIO,R0
15932  061634  005003                       CLR     R3
15933
15934  061636  172420               J2:     LDD     (R0)+,AC0       ;TEST INSTRUCTION
15935  061640  005203               J3:     INC     R3
15936  061642  005203               J4:     INC     R3
15937
15938  061644  012701  061740               MOV     #JDATOO,R1
15939  061650  174011                       STD     AC0,(R1)        ;PICK UP RESULTS
15940
15941  061652  020027  061720               CMP     R0,#JBUF0       ;WAS AN AUTO
15942  061656  001001                       BNE     1$
  (2)  061660  104000                       EMT                     ;
15943  061662  012702  061730       1$:     MOV     #JDATIO,R2      ;IS DATA CORRECT?
15944  061666  012703  061740               MOV     #JDATOO,R3
15945  061672  012704  000004               MOV     #4,R4
15946  061676  022223               J5:     CMP     (R2)+,(R3)+
15947  061700  001401                       BEQ     J6
  (2)  061702  104000                       EMT                     ;
15948  061704  077404               J6:     SOB     R4,J5
15949
15950  061706  022700  061740               CMP     #JDATIO+10,R0   ;WAS R0 INCREM.
15951  061712  001401                       BEQ     J7
  (2)  061714  104000                       EMT                     ;
15952  061716  000420               J7:     BR      JDONE
15953
15954  061720  010421               JBUF0:  .WORD   010421
15955  061722  021042               JBUF1:          021042
15956  061724  042104               JBUF2:          042104
15957  061726  031463               JBUF3:          031463
15958
15959  061730  052525               JDATIO:         052525
15960  061732  114631               JDATI1:         114631
15961  061734  063146               JDATI2:         063146
15962  061736  073567               JDATI3:         073567
15963
15964  061740  000000               JDATOO:         0
15965  061742  000000               JDATO1:         0
15966  061744  000000               JDATO2:         0
15967  061746  000000               JDATO3:         0
```

```
15968
15969  061750  177777                JDAT0:          -1
15970  061752  177777                JDAT1:          -1
15971  061754  177777                JDAT2:          -1
15972  061756  177777                JDAT3:          -1
15973
15974
15975  061760                        JDONE:
  (1)  061760  004767  042566              JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                               ;SEE IF THE USER HAS EXPRESSED
  (1)                                                               ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                               ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                               ;THE USER TYPED CONTROL G?).
15976
15983
15984                               ;*****************************************************************************
  (2)                               ;TEST 446       FSRC MODE 4 TEST
  (3)                               ;*****************************************************************************
  (2)  061764                       TS446:
15985  061764  170011                      SETD                    ;SET DOUBLE MODE
15986
15987  061766  012700  062114              MOV     #KPAT0,R0
15988  061772  172410                      LDD     (R0),AC0        ;LOAD A DEFAULT
15989                                                               ;PATTERN INTO AC0
15990  061774  012700  062074              MOV     #KBUF0,R0
15991  062000  005003                      CLR     R3
15992  062002  172440                KX2:  LDD     -(R0),AC0       ;TEST INSTRUCTION
15993  062004  005203                KX3:  INC     R3
15994  062006  005203                KX4:  INC     R3
15995
15996  062010  012701  062104              MOV     #KDAT00,R1
15997  062014  174011                      STD     AC0,(R1)        ;PICK UP THE RESULT
15998
15999  062016  020027  062104              CMP     R0,#KBUF0+10    ;WAS AN AUTO
16000  062022  001001                      BNE     1$
  (2)  062024  104000                      EMT                     ;
16001  062026  012702  062064        1$:   MOV     #KDATI0,R2      ;IS DATA CORRECT?
16002  062032  012703  062104              MOV     #KDAT00,R3
16003  062036  012704  000004              MOV     #4,R4
16004  062042  022223                KX5:  CMP     (R2)+,(R3)+
16005  062044  001401                      BEQ     KX6
  (2)  062046  104000                      EMT                     ;
16006  062050  077404                KX6:  SOB     R4,KX5
16007
16008  062052  022700  062064              CMP     #KBUF0-10,R0    ;WAS R0 DECREMENTED
16009  062056  001401                      BEQ     KX7
  (2)  062060  104000                      EMT                     ;
16010  062062  000420                KX7:  BR      KDONE
16011
16012  062064  052525                KDATI0: .WORD   052525
16013  062066  114631                KDATI1:         114631
16014  062070  063140                KDATI2:         063140
16015  062072  073567                KDATI3:         073567
16016
16017  062074  010421                KBUF0:          010421
16018  062076  031463                KBUF1:          031463
```

```
16019  062100  042104                  KBUF2:          042104
16020  062102  021042                  KBUF3:          021042
16021
16022  062104  000000                  KDAT00:         0
16023  062106  000000                  KDAT01:         0
16024  062110  000000                  KDAT02:         0
16025  062112  000000                  KDAT03:         0
16026
16027  062114  177777                  KPAT0:          -1
16028  062116  177777                  KPAT1:          -1
16029  062120  177777                  KPAT2:          -1
16030  062122  177777                  DPAT3:          -1
16031
16032  062124                          KDONE:
  (1)  062124  004767  042422                JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                 ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                 ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                 ;THE USER TYPED CONTROL G?).
16033
16040
16041                           ;*****************************************************************************
  (2)                           ;TEST 447       FSRC MODE 2, WITH FD=0, TEST
  (3)                           ;*****************************************************************************
  (2)  062130                   TS447:
16042  062130                   L1:
16043  062130  170011                    SETD                    ;SET DOUBLE MODE
16044
16045  062132  012700  062256             MOV     #LPAT10,R0
16046  062136  172410                     LDD     (R0),AC0        ;LOAD AC0
16047
16048  062140  012700  062300             MOV     #LDATI0,R0      ;SET UP THE INPUT
16049  062144  012701  062266             MOV     #LPAT20,R1      ;DATA
16050  062150  012702  000004             MOV     #4,R2
16051
16052  062154  012120                1$:  MOV     (R1)+,(R0)+
16053  062156  077202                     SOB     R2,1$
16054
16055  062160  012700  062300             MOV     #LDATI0,R0
16056  062164  005003                     CLR     R3
16057  062166  170001                     SETF                    ;CLEAR FD.
16058
16059  062170  172420                L2:  LDF     (R0)+,AC0
16060  062172  005203                L3:  INC     R3
16061
16062  062174                          L4:
16063  062174  170011                    SETD                    ;SET FD
16064
16065  062176  012701  062312             MOV     #LDAT00,R1
16066  062202  174011                     STD     AC0,(R1)        ;PICK UP RESULTS
16067
16068  062204  020027  062304             CMP     R0,#LDATI2      ;WAS R0 INCREMENTED
16069  062210  001401                     BEQ     1$
  (2)  062212  104000                     EMT                     ;
16070  062214  012737  177777  062304 1$:  MOV     #-1,@#LDATI2
16071  062222  012737  177777  062306       MOV     #-1,@#LDATI3
```

```
16072  062230  012702  062300                MOV     #LDATI0,R2      ;IS DATA CORRECT
16073  062234  012703  062312                MOV     #LDATO0,R3
16074  062240  012704  000004                MOV     #4,R4
16075
16076  062244  022223              L5:       CMP     (R2)+,(R3)+
16077  062246  001401                        BEQ     L6
  (2)  062250  104000                        EMT                     ;
16078  062252  077404              L6:       SOB     R4,L5
16079  062254  000422                        BR      LDONE
16080
16081  062256  177777              LPAT10:   .WORD   -1
16082  062260  177777              LPAT11:           -1
16083  062262  177777              LPAT12:           -1
16084  062264  177777              LPAT13:           -1
16085
16086  062266  052525              LPAT20:           052525
16087  062270  114631              LPAT21:           114631
16088  062272  063142              LPAT22:           063142
16089  062274  073567              LPAT23:           073567
16090  062276  000001                        .WORD   000001
16091  062300  000000              LDATI0:           0
16092  062302  000000              LDATI1:           0
16093  062304  000000              LDATI2:           0
16094  062306  000000              LDATI3:           0
16095  062310  000001                        .WORD   00001
16096  062312  000000              LDATO0:           0
16097  062314  000000              LDATO1:           0
16098  062316  000000              LDATO2:           0
16099  062320  000000              LDATO3:           0
16100
16101  062322                      LDONE:
  (1)  062322  004767  042224                JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                 ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                 ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                 ;THE USER TYPED CONTROL G?).
16102
16110
16111                             ;*************************************************************************************
  (2)                             ;TEST 450       FSRC MODE 2 WITH GR7, IMMEDIATE MODE, TEST
  (3)                             ;*************************************************************************************
  (2)  062326                     TS450:
16112
16113  062326                     M1:
16114  062326  170011                        SETD
16115
16116  062330  012700  062430                MOV     #MPAT10,R0
16117  062334  172410                        LDD     (R0),AC0        ;LOAD BACKROUND
16118                                                                 ;PATTERN INTO AC0.
16119  062336  005004                        CLR     R4
16120  062340  012737  062366  000004        MOV     #MERR3,@#ERRVECT
16121
16122  062346  172427  000000      M15:      LDD     #0,AC0          ;TEST INSTRUCTION
16123          062350                        .=.-2           ;EFFECTIVELY: 05204 IS PUT IN THE FIRST
16124  062350  005204                        .WORD   5204    ;16 BIT WORD, OR THE "EXP-FRACTION" WORD.
16125  062352  005204              M2:       INC     R4              ;NOTE THAT
```

```
16126  062354  005204                  M3:     INC     R4              ;005204=INC R4
16127  062356  005204                  M4:     INC     R4
16128  062360  020427  000003                  CMP     R4,#3           ;SEE IF THE PC
16129  062364  001401                          BEQ     M8
  (1)  062366                          MERR3:
  (2)  062366  104000                          EMT                     ;
16130  062370  012700  062450          M8:     MOV     #MDAT00,R0
16131  062374  174010                          STD     AC0,(R0)        ;GET THE DATA
16132
16133  062376  012700  062450                  MOV     #MDAT00,R0
16134  062402  022720  005204                  CMP     #5204,(R0)+     ;IS THE DATA CORRECT?
16135  062406  001401                          BEQ     M5
  (2)  062410  104000                          EMT                     ;
16136  062412  012701  000003          M5:     MOV     #3,R1
16137  062416  005720                  M6:     TST     (R0)+
16138  062420  001401                          BEQ     M7
  (2)  062422  104000                          EMT                     ;
16139  062424  077104                  M7:     SOB     R1,M6
16140  062426  000414                          BR      MDONE
16141
16142  062430  177777                  MPAT10:         -1
16143  062432  177777                  MPAT11:         -1
16144  062434  177777                  MPAT12:         -1
16145  062436  177777                  MPAT13:         -1
16146
16147  062440  005204                  MPAT20:         5204
16148  062442  005204                  MPAT21:         5204
16149  062444  005204                  MPAT22:         5204
16150  062446  005204                  MPAT23:         5204
16151
16152  062450  000000                  MDAT00:         0
16153  062452  000000                  MDAT01:         0
16154  062454  000000                  MDAT02:         0
16155  062456  000000                  MDAT03:         0
16156
16157  062460                          MDONE:
  (1)  062460  004767  042066                  JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
16158
16165
16166                                  ;*************************************************************************************
  (2)                                  ;TEST 451       FSRC MODE 3 TEST
  (3)                                  ;*************************************************************************************
  (2)  062464                          TS451:
16167
16168  062464                          N1:
16169  062464  170011                          SETD                    ;SET FD MODE
16170
16171  062466  012700  062614                  MOV     #NPAT10,R0
16172  062472  172410                          LDD     (R0),AC0        ;LOAD AC0 WITH A DEFAULT
16173                                                                   ;PATTERN
16174  062474  012700  062602                  MOV     #NPAT20,R0
16175  062500  005003                          CLR     R3
```

```
16176  062502  012737  062554  000004          MOV     #NERRO,@#ERRVECT        ;IF A FAILURE OCCURS
16177                                                                           ;IN THE FSRC FLOWS AN
16178                                                                           ;ODD TRAP TO 4 COULD OCCUR
16179  062510  172430                  N2:     LDD     @(R0)+,AC0              ;TEST INSTRUCTION.
16180  062512  005203                  N3:     INC     R3
16181  062514  005203                  N4:     INC     R3
16182
16183  062516  012701  062562                  MOV     #NDAT00,R1
16184  062522  174011                  STD     AC0,(R1)                ;GET THE DATA
16185
16186  062524  020027  062604                  CMP     R0,#NPAT20+2            ;WAS R0 INCREMENTED
16187  062530  001401                  BEQ     N12
  (2)  062532  104000                  EMT                             ;
16188  062534  012702  062562          N12:    MOV     #NDAT00,R2              ;DATA CORRECT
16189  062540  012703  062624                  MOV     #NDATI0,R3
16190  062544  012704  000004                  MOV     #4,R4
16191  062550  022223                  N13:    CMP     (R2)+,(R3)+
16192  062552  001401                  BEQ     N14
  (1)  062554                          NERRO:
  (2)  062554  104000                  EMT                             ;
16193  062556  077404                  N14:    SOB     R4,N13
16194  062560  000425                  BR      NDONE
15195
16196  062562  000000                  NDAT00: .WORD   0
16197  062564  000000                  NDAT01:         0
16198  062566  000000                  NDAT02:         0
16199  062570  000000                  NDAT03:         0
16200
16201  062572  052525  052525  052525          .WORD   52525,52525,52525,52525
       062600  052525
16202  062602  062624                  NPAT20: .WORD   NDATI0
16203  062604  070707                  NPAT21:         070707
16204  062606  070707                  NPAT22:         070707
16205  062610  070707                  NPAT23:         070707
16206  062612  000001                          .WORD   1
16207  062614  177777                  NPAT10: .WORD   -1
16208  062616  177777                  NPAT11:         -1
16209  062620  177777                  NPAT12:         -1
16210  062622  177777                  NPAT13:         -1
16211
16212  062624  010421                  NDATI0: .WORD   010421
16213  062626  021042                  NDATI1:         021042
16214  062630  031463                  NDATI2:         031463
16215  062632  042104                  NDATI3:         042104
16216
16217  062634                          NDONE:
  (1)  062634  004767  041712                  JSR     PC,..RSET               ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                           ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                           ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                           ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                           ;THE USER TYPED CONTROL G?).
16218
16225
16226                           ;*************************************************************************************
  (2)                           ;TEST 452       FSRC MODE 5 TEST
  (3)                           ;*************************************************************************************
```

```
  (2) 062640                          TS452:
16227
16228 062640                          01:
16229 062640 170011                           SETD                    ;SET FD MODE
16230
16231 062642 012700 062770                    MOV     #OPAT10,R0
16232 062646 172410                           LDD     (R0),AC0        ;LOAD AC0 WITH A
16233                                                                 ;DEFAULT PATTERN.
16234 062650 012700 062756                    MOV     #OPAT21,R0
16235 062654 005003                           CLR     R3
16236 062656 012737 062706 000004             MOV     #OERR0,@#ERRVEC ;IF A FAILURE
16237                                                                 ;OCCURS IN THE FSRC
16238                                                                 ;FLOWS AN ODD ADDR.
16239                                                                 ;TRAP TO 4 MAY OCCUR.
16240 062664 172450                   02:     LDD     @-(R0),AC0      ;TEST INSTRUCTION
16241 062666 005203                   03:     INC     R3
16242 062670 005203                   04:     INC     R3
16243
16244 062672 012701 062736                    MOV     #ODAT00,R1
16245 062676 174011                           STD     AC0,(R1)        ;GET THE DATA
16246
16247 062700 020027 062754                    CMP     R0,#OPAT20      ;WAS R0 DECREMENTED
16248 062704 001401                           BEQ     012
  (1) 062706                          OERR0:
  (2) 062706 104000                           EMT                     ;
16249 062710 012702 062736            012:    MOV     #ODAT00,R2      ;DATA CORRECT?
16250 062714 012703 063000                    MOV     #ODATI0,R3
16251 062720 012704 000004                    MOV     #4,R4
16252 062724 022223                   013:    CMP     (R2)+,(R3)+
16253 062726 001401                           BEQ     014
  (2) 062730 104000                           EMT                     ;
16254 062732 077404                   014:    SOB     R4,013
16255 062734 000425                           BR      ODONE
16256
16257 062736 000000                   ODAT00: .WORD   0
16258 062740 000000                   ODAT01:         0
16259 062742 000000                   ODAT02:         0
16260 062744 000000                   ODAT03:         0
16261
16262 062746 052525 052525 052525             .WORD   52525,52525,52525
16263 062754 063000                   OPAT20: .WORD   ODATI0
16264 062756 070707                   OPAT21: 070707
16265 062760 070707                   OPAT22: 070707
16266 062762 070707                   OPAT23: 070707
16267 062764 070707                   OPAT24: 070707
16268 062766 000001                           .WORD   1
16269 062770 177777                   OPAT10: .WORD   -1
16270 062772 177777                   OPAT11:         -1
16271 062774 177777                   OPAT12:         -1
16272 062776 177777                   OPAT13:         -1
16273
16274 063000 073567                   ODATI0: .WORD   73567
16275 063002 004210                   ODATI1:         004210
16276 063004 114631                   ODATI2:         114631
16277 063006 125252                   ODATI3:         125252
16278
```

```
16279  063010                          ODONE:
  (1)  063010  004767  041536                JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                 ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                 ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                 ;THE USER TYPED CONTROL G?).
16280
16286
16287                             ;*************************************************************************************
  (2)                             ;TEST 453       FSRC MODE 6 TEST
  (3)                             ;*************************************************************************************
  (2)  063014                     TS453:
16288
16289  063014                     P1:
16290  063014  170011                      SETD                    ;SET FD MODE
16291
16292  063016  012700  063102              MOV     #PPAT1O,R0
16293  063022  172410                      LDD     (R0),AC0        ;LOAD A DEFAULT PATTERN
16294                                                               ;INTO AC0
16295  063024  012700  062651              MOV     #PDATI0-241,R0  ;COULD OCCUR.
16296
16297  063030  172460  000241     P2:      LDD     241(R0),AC0
16298          063032             P3=P2+2
16299
16300  063034  012701  063122     P4:      MOV     #PDAT00,R1
16301  063040  174011                      STD     AC0,(R1)        ;GET THE DATA
16302  063042  012703  000004              MOV     #4,R3
16303  063046  012702  063112              MOV     #PDATI0,R2
16304  063052  012701  063122              MOV     #PDAT00,R1
16305  063056  022221             P5:      CMP     (R2)+,(R1)+     ;CHECK THE DATA
16306  063060  001401                      BEQ     2$
  (2)  063062  104000                      EMT                     ;
16307  063064  077304             2$:      SOB     R3,P5
16308  063066  022700  062651              CMP     #PDATI0-241,R0  ;R0 CORRECT?
16309  063072  001401                      BEQ     1$
  (2)  063074  104000                      EMT                     ;
16310  063076  000137  063132     1$:      JMP     @#PDONE
16311  063102  177777             PPAT10:  .WORD   -1
16312  063104  177777             PPAT11:          -1
16313  063106  177777             PPAT12:          -1
16314  063110  177777             PPAT13:          -1
16315
16316  063112  010421             PDATI0:  .WORD   010421
16317  063114  031463             PDATI1:          031463
16318  063116  052525             PDATI2:          052525
16319  063120  073567             PDATI3:          073567
16320
16321  063122  000000             PDAT00:  .WORD   0
16322  063124  000000             PDAT01:          0
16323  063126  000000             PDAT02:          0
16324  063130  000000             PDAT03:          0
16325
16326  063132                     PDONE:
  (1)  063132  004767  041414                JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                 ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                 ;THE DESIRE TO CHANGE THE SOFTWARE
```

```
  (1)                                                   ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                   ;THE USER TYPED CONTROL G?).
16327
16334
16335                           ;^*************************************************************************************
  (2)                           ;TEST 454       FSRC MODE 7 TEST
  (3)                           ;*************************************************************************************
  (2) 063136                    TS454:
16336
16337 063136                    Q1:
16338 063136 170011                     SETD
16339
16340 063140 012700 063224              MOV     #QPAT10,R0
16341 063144 172410                     LDD     (R0),AC0        ;LOAD A DEFAULT
16342                                                           ;PATTERN INTO AC0
16343 063146 012700 062773              MOV     #QPAT20-241,R0
16344
16345 063152 172470 000241      Q2:     LDD     @241(R0),AC0
16346        063154             Q3=Q2+2
16347
16348 063156 012701 063244      Q4:     MOV     #QDAT00,R1
16349 063162 174011                     STD     AC0,(R1)        ;GET THE DATA
16350
16351 063164 012703 000004              MOV     #4,R3
16352 063170 012704 063244              MOV     #QDAT00,R4
16353 063174 012705 063254              MOV     #QDATI0,R5
16354 063200 022425             Q5:     CMP     (R4)+,(R5)+     ;CHECK THE DATA
16355 063202 001401                     BEQ     2$
  (2) 063204 104000                     EMT                     ;
16356 063206 077304             2$:     SOB     R3,Q5
16357
16358 063210 022700 062773              CMP     #QPAT20-241,R0  ;CHECK R0.
16359 063214 001401                     BEQ     1$
  (2) 063216 104000                     EMT                     ;
16360 063220 000137 063264      1$:     JMP     @#QDONE
16361
16362 063224 177777             QPAT10: .WORD   -1
16363 063226 177777             QPAT11:         -1
16364 063230 177777             QPAT12:         -1
16365 063232 177777             QPAT13:         -1
16366
16367 063234 063254             QPAT20: .WORD   QDATI0
16368 063236 052525             QPAT21:         52525
16369 063240 052525             QPAT22:         52525
16370 063242 052525             QPAT23:         52525
16371
16372 063244 000000             QDAT00: .WORD   0
16373 063246 000000             QDAT01:         0
16374 063250 000000             QDAT02:         0
16375 063252 000000             QDAT03:         0
16376
16377 063254 073567             QDATI0: .WORD   073567
16378 063256 052525             QDATI1: .WORD   052525
16379 063260 031463             QDATI2: .WORD   031463
16380 063262 010421             QDATI3: .WORD   010421
16381
```

```
16382  063264                          QDONE:
  (1)  063264  004767  041262                JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                 ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                 ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                 ;THE USER TYPED CONTROL G?).
16383
16398                                 ;*************************************************************************
  (2)                                 ;TEST 455       (BUT EZBT Y8),(BUT ENBT) AND (BUT FIUV) TEST
  (3)                                 ;*************************************************************************
  (2)  063270                         TS455:
16399  063270  005037  064042                CLR     @#UFLAG
16400  063274  012700  063772                MOV     #UPAT00,R0      ;SET UP AC#0 DATA.
16401  063300  012701  000004                MOV     #4,R1
16402  063304  012720  177777          U0:   MOV     #-1,(R0)+
16403  063310  077103                        SOB     R1,U0
16404
16405  063312  012737  000033  064044        MOV     #033,@#UTMP1
16406  063320  012737  000023  064046        MOV     #023,@#UTMP2
16407  063326  012737  063412  000244  U1:   MOV     #UERR0,@#FPVECT ;IN CASE (BUT FIUV FAILS)
16408  063334  012700  000200                MOV     #200,R0
16409  063340  170100                        LDFPS   R0
16410  063342  012700  063772                MOV     #UPAT00,R0      ;LOAD AC0
16411  063346  172410                        LDD     (R0),AC0
16412  063350  013737  064044  064050        MOV     @#UTMP1,@#UROM1
16413  063356  012737  000001  064052        MOV     #001,@#UROM2
16414  063364  012737  000254  064054        MOV     #254,@#UROM3
16415
16416  063372  012700  064002                MOV     #UPAT10,R0      ;LOAD 0 INTO AC0
16417  063376  172410                  U2:   LDD     (R0),AC0
16418  063400  012704  000204                MOV     #204,R4         ;SEE IF FPS IS CORRECT
16419  063404  170205                        STFPS   R5
16420
16421  063406  020405                        CMP     R4,R5
16422  063410  001401                        BEQ     U3
  (1)  063412                         UERR0:
  (2)  063412  104000                        EMT                     ;
16423  063414  012700  000200          U3:   MOV     #200,R0
16424  063420  170100                        LDFPS   R0
16425
16426  063422  012700  063772                MOV     #UPAT00,R0      ;LOAD AC0
16427  063426  172410                        LDD     (R0),AC0
16428  063430  013737  064046  064050        MOV     @#UTMP2,@#UROM1
16429  063436  012737  000003  064052        MOV     #003,@#UROM2
16430  063444  012737  000054  064054        MOV     #054,@#UROM3
16431
16432  063452  012700  064012                MOV     #UPAT20,R0      ;LOAD A POSITIVE NUMBER
16433                                                                 ;INTO AC0
16434  063456  172410                  U4:   LDD     (R0),AC0
16435  063460  012704  000200                MOV     #200,R4         ;FPS CORRECT?
16436  063464  170205                        STFPS   R5
16437
16438  063466  020405                        CMP     R4,R5
16439  063470  001401                        BEQ     U5
  (2)  063472  104000                        EMT                     ;
16440  063474  012700  000200          U5:   MOV     #200,R0
```

```
16441  063500  170100                        LDFPS   R0
16442  063502  012700  063772                MOV     @UPAT00,R0      ;LOAD AC0
16443  063506  172410                        LDD     (R0),AC0
16444  063510  013737  064046  064050        MOV     @#UTMP2,@#UROM1
16445  063516  012737  000403  064052        MOV     #403,@#UROM2
16446  063524  012737  000056  064054        MOV     #056,@#UROM3
16447  063532  012700  064022                MOV     @UPAT30,R0      ;LOAD A NEGATIUE
16448                                                                ;NUMBER INTO AL0
16449  063536  172410                U6:     LDD     (R0),AC0
16450  063540  012704  000210                MOV     #210,R4         ;FPS CORRECT
16451  063544  170205                        STFPS   R5
16452  063546  020405                        CMP     R4,R5
16453  063550  001401                        BEQ     U7
  (2)  063552  104000                        EMT                     ;
16454  063554  012700  000200        U7:     MOV     #200,R0
16455  063560  170100                        LDFPS   R0
16456  063562  012700  063772                MOV     @UPAT00,R0      ;LOAD AC0
16457  063566  172410                        LDD     (R0),AC0
16458  063570  013737  064044  064050        MOV     @#UTMP1,@#UROM1
16459  063576  012737  000401  064052        MOV     #401,@#UROM2
16460  063604  012737  000256  064054        MOV     #256,@#UROM3
16461  063612  012700  064032                MOV     @UPAT40,R0      ;LOAD -0 INTO AC0
16462  063616  172410                U10:    LDD     (R0),AC0
16463  063620  000240                U11:    NOP                     ;TRAP FROM HERE IF
16464  063622  012704  000214                MOV     #214,R4         ;SEE IF FPS IS CORRECT.
16465  063626  170205                        STFPS   R5
16466  063630  020405                        CMP     R4,R5
16467  063632  001401                        BEQ     U12
  (2)  063634  104000                        EMT                     ;
16468  063636  005737  064042        U12:    TST     @#UFLAG ;SEE IF ALL THE PATTERNS
16469  063642  001021                        BNE     U14             ;HAVE BEEN TEST WITH
16470                                                                ;BOTH AC NOT EQUAL TO 0 AND AC=0
16471  063644  012700  063772                MOV     @UPAT00,R0      ;IF NOT GO BACK AND
16472  063650  012701  000004                MOV     #4,R1           ;CHECK THEM WITH AC=0
16473  063654  005020                U13:    CLR     (R0)+
16474  063656  077102                        SOB     R1,U13
16475  063660  012737  177777  064042        MOV     #-1,@#UFLAG
16476  063666  012737  000233  064044        MOV     #233,@#UTMP1
16477  063674  012737  000223  064046        MOV     #223,@#UTMP2
16478  063702  000137  063326                JMP     @#U1
16479                        ;NOW SEE IF A TRAP CAN BE FORCED BY SETTING FIUV AND LOADING -0
16480  063706  012737  063744  000244  U14:    MOV     @UERR3,@#FPVECT
16481  063714  012700  004200                MOV     #4200,R0        ;SET FD AND FIUV
16482  063720  170100                        LDFPS   R0
16483  063722  012700  063772                MOV     @UPAT00,R0      ;SET UP AC0
16484  063726  172410                        LDD     (R0),AC0
16485  063730  012700  064032                MOV     @UPAT40,R0      ;LOAD -0
16486  063734  172410                U15:    LDD     (R0),AC0        ;SHOULD TRAP TO 244
16487  063736  170000                U16:    CFCC
16488  063740  000240                        NOP
16489  063742  104000                        EMT                     ;
16490
16491                        ;INTERRUPT HERE WHEN FIUV SET AND ATTEMPTED TO LOAD-0
16492  063744  021627  063736        UERR3:  CMP     (SP),@U16
16493  063750  001401                        BEQ     1$
  (2)  063752  104000                        EMT                     ;
```

```
16494  063754  022626              1$:     CMP     (SP)+,(SP)+
16495  063756  005000                      CLR     R0
16496  063760  170300                      STST    R0              ;GET FEC.
16497  063762  022700  000014              CMP     #14,R0          ;CORRECT
16498  063766  001433                      BEQ     UDONE
  (2)  063770  104000                      EMT                     ;
16499  063772  000000              UPAT00: .WORD   0
16500  063774  000000              UPAT01:         0
16501  063776  000000              UPAT02:         0
16502  064000  000000              UPAT03:         0
16503
16504  064002  000000              UPAT10: .WORD   0               ;0
16505  064004  000000              UPAT11:         0
16506  064006  000000              UPAT12:         0
16507  064010  000000              UPAT13:         0
16508
16509  064012  010421              UPAT20: .WORD   010421          ;POS NUM
16510  064014  114631              UPAT21:         114631
16511  064016  125252              UPAT22:         125252
16512  064020  177777              UPAT23:         177777
16513
16514  064022  114631              UPAT30:         114631          ;NEG NUM
16515  064024  135673              UPAT31:         135673
16516  064026  146314              UPAT32:         146314
16517  064030  167356              UPAT33:         167356
16518
16519  064032  100000              UPAT40:         100000          ;NEG ZERO
16520  064034  000000              UPAT41:         0
16521  064036  000000              UPAT42:         0
16522  064040  000000              UPAT43:         0
16523
16524  064042  000000              UFLAG:  .WORD   0
16525  064044  000000              UTMP1:          0
16526  064046  000000              UTMP2:          0
16527  064050  000000              UROM1:          0
16528  064052  000000              UROM2:          0
16529  064054  000000              UROM3:          0
16530  064056                      UDONE:
16531
16532
16540                              ;*****************************************************************************
  (2)                              ;TEST 456       ADDF,ADDD,SUBF AND SUBD WITH FSRC=AC=0 TEST
  (3)                              ;*****************************************************************************
  (2)  064056                      TS456:
16541  064056  012700  000200              MOV     #200,R0
16542  064062  170100                      LDFPS   R0              ;SET DOUBLE MODE
16543  064064  012700  064406              MOV     #UPAT00,R0      ;LOAD AC0=:
16544  064070  172410                      LDD     (R0),AC0
16545  064072  012700  064406              MOV     #UPAT00,R0
16546  064076  172010              W2:     ADDD    (R0),AC0        ;TEST INSTRUCTION. ADD ITSELF
16547  064100  170205                      STFPS   R5              ;GET FPS
```

```
16549  064102  170011                        SETD                    ;SET DOUBLE MODE
16550  064104  012700  064406                MOV     #WPAT00,R0
16551  064110  174010                        STD     AC0,(R0)        ;GET THE RESULT
16552  064112  012701  064406                MOV     #WPAT00,R1
16553  064116  012702  000004                MOV     #4,R2
16554  064122  022021              W3:       CMP     (R0)+,(R1)+     ;IS RESULT CORRECT
16555  064124  001401                        BEQ     W4
  (2)  064126  104000                        EMT                     ;
16556  064130  077204              W4:       SOB     R2,W3
16557  064132  022705  000204                CMP     #204,R5         ;IS FPS CORRECT
16558  064136  001401                        BEQ     W5
  (2)  064140  104000                        EMT                     ;
16559  064142  012700  000200      W5:       MOV     #200,R0
16560  064146  170100                        LDFPS   R0              ;SET DOUBLE MODE
16561  064150  012700  064406                MOV     #WPAT00,R0      ;LOAD AC0=0
16562  064154  172410                        LDD     (R0),AC0
16563  064156  005000                        CLR     R0
16564  064160  170100                        LDFPS   R0              ;GO TO FLOATING MODE
16565  064162  012700  064406                MOV     #WPAT00,R0
16566  064166  172010              W6:       ADDF    (R0),AC0        ;TEST INSTRUCTION
16567  064170  170205                        STFPS   R5              ;GET FPS
16568  064172  170011                        SETD                    ;RESET TO DOUBLE MODE
16569  064174  012700  064406                MOV     #WPAT00,R0
16570  064200  174010                        STD     AC0,(R0)        ;GET THE RESULT
16571  064202  012701  064406                MOV     #WPAT00,R1
16572  064206  012702  000004                MOV     #4,R2
16573  064212  022021              W7:       CMP     (R0)+,(R1)+     ;WAS THE RESULT
16574  064214  001401                        BEQ     W10
  (2)  064216  104000                        EMT                     ;
16575  064220  077204              W10:      SOB     R2,W7
16576  064222  022705  000004                CMP     #4,R5           ;WAS FPS CORRECT
16577  064226  001401                        BEQ     W11
  (2)  064230  104000                        EMT                     ;
16578  064232  012700  000200      W11:      MOV     #200,R0
16579  064236  170100                        LDFPS   R0              ;SET DOUBLE MODE
16580  064240  012700  064406                MOV     #WPAT00,R0      ;LOAD AC0=0
16581  064244  172410                        LDD     (R0),AC0
16582  064246  012700  064406                MOV     #WPAT00,R0
16583  064252  173010              W12:      SUBD    (R0),AC0        ;TEST INSTRUCTION
16584  064254  170205                        STFPS   R5              ;GET FPS
16585  064256  170011                        SETD                    ;SET DOUBLE MODE
16586  064260  012700  064406                MOV     #WPAT00,R0
16587  064264  174010                        STD     AC0,(R0)        ;GET THE RESULT
16588  064266  012701  064406                MOV     #WPAT00,R1
16589  064272  012702  000004                MOV     #4,R2
16590  064276  022021              W13:      CMP     (R0)+,(R1)+     ;IS RESULT CORRECT?
16591  064300  001401                        BEQ     W14
  (2)  064302  104000                        EMT                     ;
16592  064304  077204              W14:      SOB     R2,W13
16593  064306  022705  000204                CMP     #204,R5         ;IS FPS CORRECT?
16594  064312  001401                        BEQ     W15
  (2)  064314  104000                        EMT                     ;
16595  064316  012700  000200      W15:      MOV     #200,R0
16596  064322  170100                        LDFPS   R0              ;SET DOUBLE MODE
16597  064324  012700  064406                MOV     #WPAT00,R0      ;LOAD AC0=0
16598  064330  172410                        LDD     (R0),AC0
```

```
16599  064332  005000                        CLR     R0
16600  064334  170100                        LDFPS   R0              ;ENTER FLOATING MODE.
16601  064336  012700  064406                MOV     #WPAT00,R0
16602  064342  173010                W16:   SUBF    (R0),AC0        ;TEST INSTRUCTION.
16603  064344  170205                       STFPS   R5              ;GET FPS
16604  064346  170011                       SETD                    ;RESET TO DOUBLE MODE
16605  064350  012700  064406                MOV     #WPAT00,R0      ;GET THE RESULT.
16606  064354  174010                       STD     AC0,(R0)
16607  064356  012701  064406                MOV     #WPAT00,R1
16608  064362  012702  000004                MOV     #4,R2
16609  064366  022021                W17:   CMP     (R0)+,(R1)+     ;IS RESULT CORRECT?
16610  064370  001401                       BEQ     W20
  (2)  064372  104000                       EMT                     ;
16611  064374  077204                W20:   SOB     R2,W17
16612  064376  022705  000004                CMP     #4,R5           ;IS FPS CORRECT?
16613  064402  001411                       BEQ     WDONE
  (2)  064404  104000                       EMT                     ;
16614
16615  064406  000000                WPAT00: .WORD   0
16616  064410  000000                WPAT01:         0
16617  064412  000000                WPAT02:         0
16618  064414  000000                WPAT03:         0
16619
16620  064416  000000                WDAP00: .WORD   0
16621  064420  000000                WDAT01:         0
16622  064422  000000                WDAT02:         0
16623  064424  000000                WDAT03:         0
16624
16625  064426                        WDONE:
  (1)  064426  004767  040120                JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                ;THE USER TYPED CONTROL G?).
16626
16627
16634                                 ;*****************************************************************************
  (2)                                 ;TEST 457       ADDD AND SUB WITH FSRC=0
  (3)                                 ;*****************************************************************************
  (2)  064432                         TS457:
16635  064432  012700  000200                MOV     #200,R0
16636  064436  170100                       LDFPS   R0              ;SET DOUBLE MODE
16637  064440  012700  064772                MOV     #XPAT00,R0      ;SET AC0 TO POSITIVE
16638  064444  172410                       LDD     (R0),AC0
16639  064446  012700  065002                MOV     #XPAT10,R0      ;FSRC=0
16640  064452  172010                X2:    ADDD    (R0),AC0        ;TEST INSTRUCTION
16641  064454  170205                       STFPS   R5
16642  064456  170011                       SETD
16643  064460  012700  064762                MOV     #XDAT00,R0      ;GET RESULT.
16644  064464  174010                       STD     AC0,(R0)
16645  064466  012701  064772                MOV     #XPAT00,R1
16646  064472  012702  000004                MOV     #4,R2
16647  064476  022021                X3:    CMP     (R0)+,(R1)+     ;IS RESULT CORRECT?
16648  064500  001401                       BEQ     X4
  (2)  064502  104000                       EMT                     ;
16649  064504  077204                X4:    SOB     R2,X3
```

F6



```
16650  064506  012704  000200                MOV     #200,R4
16651  064512  020405                        CMP     R4,R5           ;IS FPS CORRECT?
16652  064514  001401                        BEQ     X5
  (2)  064516  104000                        EMT                     ;
16653  064520  012700  000200          X5:   MOV     #200,R0
16654  064524  170100                        LDFPS   R0              ;SET DOUBLE MODE
16655  064526  012700  065012                MOV     #XPAT20,R0      ;SET AC0 TO
16656  064532  172410                        LDD     (R0),AC0
16657  064534  012700  065002                MOV     #XPAT10,R0      ;FSRC=0
16658  064540  172010                  X6:   ADDD    (R0),AC0        ;TEST INSTRUCTION
16659  064542  170205                        STFPS   R5
16660  064544  170011                        SETD
16661  064546  012700  064762                MOV     #XDAT00,R0      ;GET RESULT
16662  064552  174010                        STD     AC0,(R0)
16663  064554  012701  065012                MOV     #XPAT20,R1
16664  064560  012702  000004                MOV     #4,R2
16665  064564  022021                  X7:   CMP     (R0)+,(R1)+     ;IS RESULT CORRECT?
16666  064566  001401                        BEQ     X10
  (2)  064570  104000                        EMT                     ;
16667  064572  077204                  X10:  SOB     R2,X7
16668  064574  012704  000210                MOV     #210,R4
16669  064600  020405                        CMP     R4,R5           ;IS FPS CORRECT?
16670  064602  001401                        BEQ     X11
  (2)  064604  104000                        EMT                     ;
16671  064606  012700  000200          X11:  MOV     #200,R0
16672  064612  170100                        LDFPS   R0              ;SET DOUBLE MODE
16673  064614  012700  064772                MOV     #XPAT00,R0      ;SET AC0 TO NON-ZERO
16674  064620  172410                        LDD     (R0),AC0
16675  064622  012700  065002                MOV     #XPAT10,R0      ;FSRC=0
16676  064626  173010                  X12:  SUBD    (R0),AC0        ;TEST INSTRUCTION
16677  064630  170205                        STFPS   R5
16678  064632  170011                        SETD
16679  064634  012700  064762                MOV     #XDAT00,R0      ;GET RESULT
16680  064640  174010                        STD     AC0,(R0)
16681  064642  012701  064772                MOV     #XPAT00,R1
16682  064646  012702  000004                MOV     #4,R2
16683  064652  022021            X13:  CMP     (R0)+,(R1)+     ;IS RESULT CORRECT?
16684  064654  001401                        BEQ     X14
  (2)  064656  104000                        EMT                     ;
16685  064660  077204            X14:  SOB     R2,X13
16686  064662  012704  000200                MOV     #200,R4         ;IS FPS CORRECT?
16687  064666  020405                        CMP     R4,R5
16688  064670  001401                        BEQ     X15
  (2)  064672  104000                        EMT                     ;
16689  064674  012700  000200          X15:  MOV     #200,R0
16690  064700  170100                        LDFPS   R0              ;SET DOUBLE MODE
16691  064702  012700  065012                MOV     #XPAT20,R0      ;SET AC0=A NEGATIVE
16692  064706  172410                        LDD     (R0),AC0
16693  064710  012700  065002                MOV     #XPAT10,R0      ;FSRC=0
16694  064714  173010            X16:  SUBD    (R0),AC0        ;TEST INSTRUCTION.
16695  064716  170205                        STFPS   R5
16696  064720  170011                        SETD
16697  064722  012700  064762                MOV     #XDAT00,R0      ;GET RESULT
16698  064726  174010                        STD     AC0,(R0)
16699  064730  012701  065012                MOV     #XPAT20,R1
16700  064734  012702  000004                MOV     #4,R2
```

```
16701  064740  022021                  X17:    CMP     (R0)+,(R1)+     ;IS RESULT CORRECT?
16702  064742  001401                          BEQ     X20
  (2)  064744  104000                          EMT                     ;
16703  064746  077204                  X20:    SOB     R2,X17
16704  064750  012704  000210                  MOV     #210,R4         ;IS FPS CORRECT?
16705  064754  020405                          CMP     R4,R5
16706  064756  001421                          BEQ     XDONE
  (2)  064760  104000                          EMT                     ;
16707  064762  000000                  XDAT00: .WORD   0
16708  064764  000000                  XDAT01:         0
16709  064766  000000                  XDAT02:         0
16710  064770  000000                  XDAT03:         0
16711
16712  064772  010421                  XPAT00: .WORD   010421
16713  064774  021042                  XPAT01:         021042
16714  064776  031463                  XPAT02:         031463
16715  065000  042104                  XPAT03:         042104
16716
16717  065002  000000                  XPAT10: .WORD   0
16718  065004  000000                  XAPT11:         0
16719  065006  000000                  XPAT12:         0
16720  065010  000000                  XPAT13:         0
16721  065012  104210                  XPAT20: .WORD   104210
16722  065014  114631                  XPAT21:         114631
16723  065016  125252                  XPAT22:         125252
16724  065020  135673                  XPAT23:         135673
16725
16726  065022                          XDONE:
  (1)  065022  004767  037524                JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                 ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                 ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                 ;THE USER TYPED CONTROL G?).
16727
16735                                   ;*****************************************************************************
  (2)                                   ;TEST 460       SUBD WITH AC=0 TEST
  (3)                                   ;*****************************************************************************
  (2)  065026                           TS460:
16736  065026  005037  065202                CLR     @#YFLAG
16737  065032  012737  065222  065204        MOV     #YPAT00,@#YTMP1 ;P
16738  065040  012737  065232  065206        MOV     #YPAT10,@#YTMP2 ;N
16739  065046  012737  000210  065210        MOV     #210,@#YTMP3
16740  065054  012700  000200          Y1:   MOV     #200,R0
16741  065060  170100                        LDFPS   R0              ;SET DOUBLE MODE
16742  065062  012700  065242                MOV     #YPAT20,R0      ;SET AC0=0
16743  065066  172410                        LDD     (R0),AC0
16744  065070  013700  065204                MOV     @#YTMP1,R0
16745  065074  173010          Y2:   SUBD    (R0),AC0        ;TEST INSTRUCTION
16746  065076  170205                        STFPS   R5
16747  065100  170011                        SETD
16748  065102  012700  065212                MOV     #YDAT00,R0      ;GET RESULT
16749  065106  174010                        STD     AC0,(R0)
16750  065110  012702  000004                MOV     #4,R2
16751  065114  013701  065206                MOV     @#YTMP2,R1      ;CHECK RESULT.
16752  065120  022021          Y3:   CMP     (R0)+,(R1)+
16753  065122  001401                        BEQ     1$
```

H6



```
  (2)  065124  104000                          EMT                     ;
16754  065126  077204                  1$:     SOB     R2,Y3
16755  065130  023705  065210                  CMP     @#YTMP3,R5      ;FPS CORRECT?
16756  065134  001401                          BEQ     Y$
  (2)  065136  104000                          EMT                     ;
16757  065140  005737  065202          Y4:     TST     @#YFLAG         ;FINISHED TEST?
16758  065144  001015                          BNE     Y5
16759  065146  012737  177777  065202          MOV     #-1,@#YFLAG
16760  065154  012737  065232  065204          MOV     #YPAT10,@#YTMP1
16761  065162  012737  065222  065206          MOV     #YPAT00,@#YTMP2
16762  065170  012737  000200  065210          MOV     #200,@#YTMP3
16763  065176  000726                          BR      Y1
16764  065200  000424                  Y5:     BR      YDONE
16765
16766  065202  000000                  YFLAG:  .WORD   0
16767  065204  000000                  YTMP1:          0
16768  065206  000000                  YTMP2:          0
16769  065210  000000                  YTMP3:          0
16770
16771  065212  000000                  YDAT00: .WORD   0
16772  065214  000000                  YDAT01:         0
16773  065216  000000                  YDAT02:         0
16774  065220  000000                  YDAT03:         0
16775
16776  065222  063146                  YPAT00:         063146
16777  065224  052525                  YPAT01:         052525
16778  065226  042104                  YPAT02:         042104
16779  065230  167356                  YPAT03:         167356
16780
16781  065232  163146                  YPAT10:         163146
16782  065234  052525                  YPAT11:         052525
16783  065236  042104                  YPAT12:         042104
16784  065240  167356                  YPAT13:         167356
16785
16786  065242  000000                  YPAT20:         0
16787  065244  000000                  YPAT21:         0
16788  065246  000000                  YPAT22:         0
16789  065250  000000                  YPAT23:         0
16790
16791  065252                          YDONE:
  (1)  065252  004767  037274                  JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
16800                                  ;*******************************************************************************
  (2)                                  ;TEST 461       ADDD WITH AC=0 TEST
  (3)                                  ;*******************************************************************************
  (2)  065256                          TS461:
16801  065256  005067  000134                  CLR     ZFLAG
16802  065262  012737  065434  065420          MOV     #ZPAT00,@#ZTMP1 ;P
16803  065270  012737  000200  065422          MOV     #200,@#ZTMP2
16804  065276  012700  000200          Z1:     MOV     #200,R0
16805  065302  170100                          LDFPS   R0              ;SET DOUBLE MODE
16806  065304  012700  065454                  MOV     #ZPAT20,R0      ;SET AC0=0
16807  065310  172410                          LDD     (R0),AC0
```

```
16808  065312  013700  065420          MOV     @#ZTMP1,R0
16809  065316  172010          Z2:     ADDD    (R0),AC0        ;TEST INSTRUCTION
16810  065320  170205                  STFPS   R5
16811  065322  170011                  SETD
16812  065324  012700  065424          MOV     #ZDAT00,R0      ;GET RESULT
16813  065330  174010                  STD     AC0,(R0)
16814  065332  012702  000004          MOV     #4,R2
16815  065336  013701  065420          MOV     @#ZTMP1,R1      ;RESULT CORRECT?
16816  065342  022021          Z3:     CMP     (R0)+,(R1)+
16817  065344  001401                  BEQ     Z4
  (2)  065346  104000                  EMT                     ;
16818  065350  077204          Z4:     SOB     R2,Z3
16819  065352  023705  065422          CMP     @#ZTMP2,R5      ;FPS CORRECT?
16820  065356  001401                  BEQ     Z5
  (2)  065360  104000                  EMT                     ;
16821  065362  005737  065416          TST     @#ZFLAG         ;FINISHED TEST?
16822  065366  001012                  BNE     Z6
16823  065370  012737  177777  065416  MOV     #-1,@#ZFLAG
16824  065376  012737  065444  065420  MOV     #ZPAT10,@#ZTMP1
16825  065404  012737  000210  065422  MOV     #210,@#ZTMP2
16826  065412  000731                  BR      Z1
16827  065414  000423          Z6:     BR      ZDONE
16828
16829  065416  000000          ZFLAG:  .WORD   0
16830  065420  000000          ZTMP1:          0
16831  065422  000000          ZTMP2:          0
16832
16833  065424  000000          ZDAT00: .WORD   0
16834  065426  000000          ZDAT01:         0
16835  065430  000000          ZDAT02:         0
16836  065432  000000          ZDAT03:         0
16837
16838  065434  031463          ZPAT00:         031463
16839  065436  010421          ZPAT01:         010421
16840  065440  146314          ZPAT02:         146314
16841  065442  156735          ZPAT03:         156735
16842
16843  065444  156735          ZPAT10:         156735
16844  065446  167356          ZPAT11:         167356
16845  065450  135673          ZPAT12:         135673
16846  065452  146314          ZPAT13:         146314
16847
16848  065454  000000          ZPAT20:         0
16849  065456  000000          ZPAT21:         0
16850  065460  000000          ZPAT22:         0
16851  065462  000000          ZPAT23:         0
16852
16853  065464                  ZDONE:
  (1)  065464  004767  037062          JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                         ;SEE IF THE USER HAS EXPRESSED
  (1)                                                         ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                         ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                         ;THE USER TYPED CONTROL G?).
16854
16855
16863                   ;*****************************************************************************************
```

J6



```
  (2)                                   ;TEST 462       ADDF AND ADDD WITH E(AC)=E(FSRC) TEST AND (BUT FT) TEST
  (3)                                   ;*****************************************************************************
  (2)  065470                           TS462:
16864  065470  012700  003240                   MOV     #3240,R0
16865  065474  170100                           LDFPS   R0              ;SET FIU FIV FD AND FT
16866  065476  012700  065700                   MOV     #AAPAT0,R0      ;FLOWS IN TRAP WILL
16867                                                                   ;OCCUR
16868  065502  172410                           LDD     (R0),AC0        ;SET UP AC0
16869  065504  012700  065710                   MOV     #AAPAT1,R0
16870  065510  172010                   AA2:    ADDD    (R0),AC0        ;TEST INSTRUCTION
16871                                                                   ;SHOULD TRUNCATE
16872  065512  012700  065670           AA3:    MOV     #AADAT0,R0
16873  065516  174010                           STD     AC0,(R0)        ;GET THE RESULT
16874  065520  012701  065720                   MOV     #AAPAT2,R1
16875  065524  012702  000004                   MOV     #4,R2
16876  065530  022021                   AA4:    CMP     (R0)+,(R1)+     ;CORRECT?
16877  065532  001401                           BEQ     AA7
  (2)  065534  104000                           EMT                     ;
16878  065536  077204                   AA7:    SOB     R2,AA4
16879
16880                                   ;NOW TEST DOUBLE FLOATING ROUND MODE.
16881                                   ;A 1 SHOULD BE ADDED TO THE LSB ON ROUNG MODE.
16882
16883  065540  012700  003200                   MOV     #3200,R0        ;SET FD FIV FIV. FT=0
16884  065544  170100                           LDFPS   R0
16885  065546  012700  065700                   MOV     #AAPAT0,R0
16886  065552  172410                           LDD     (R0),AC0        ;SET UP AC0 OPERAND
16887  065554  012700  065710                   MOV     #AAPAT1,R0
16888  065560  172010                   AA11:   ADDD    (R0),AC0        ;TEST INSTRUCTION
16889                                                                   ;SHOULD ROUND
16890  065562  012700  065670           AA12:   MOV     #AADAT0,R0
16891  065566  174010                           STD     AC0,(R0)        ;GET THE RESULT
16892  065570  012701  065730                   MOV     #AAPAT3,R1
16893  065574  012702  000004                   MOV     #4,R2
16894  065600  022021                   AA13:   CMP     (R0)+,(R1)+     ;CORRECT?
16895  065602  001401                           BEQ     AA20
  (2)  065604  104000                           EMT                     ;
16896  065606  077204                   AA20:   SOB     R2,AA13
16897
16898                                   ;NOW TEST ADDF WITH FT=0, ROUND MODE
16899
16900  065610  012700  003200                   MOV     #3200,R0                ;FIV=1, FIV=1, FT=0
16901  065614  170100                           LDFPS   R0
16902  065616  012700  065700                   MOV     #AAPAT0,R0      ;LOAD AC0 OPERAND
16903  065622  172410                           LDD     (R0),AC0
16904  065624  170001                           SETF                    ;ENTER FLOATING MODE
16905  065626  012700  065750                   MOV     #AAPAT5,R0
16906  065632  172010                   AA22:   ADDF    (R0),AC0        ;TEST INSTRUCTION
16907                                                                   ;SHOULD ROUND
16908  065634                           AA23:
16909  065634  170011                           SETD                    ;RESET TO DOUBLE
16910                                                                   ;MODE
16911  065636  012700  065670                   MOV     #AADAT0,R0      ;GET THE RESULT
16912  065642  174010                           STD     AC0,(R0)
16913  065644  012701  065760                   MOV     #AAPAT6,R1      ;CORRECT?
16914  065650  012702  000002                   MOV     #2,R2
```

```
16915  065654  022021                  AA24:   CMP     (R0)+,(R1)+
16916  065656  001401                          BEQ     AA27
  (2)  065660  104000                          EMT                     ;
16917  065662  077204                  AA27:   SOB     R2,AA24
16918  065664  000137  065770                  JMP     @#AADONE
16919  065670  000000                  AADAT0:         0
16920  065672  000000                          0
16921  065674  000000                          0
16922  065676  000000                          0
16923  065700  000200                  AAPAT0:         200
16924  065702  000000                          0
16925  065704  000000                          0
16926  065706  000000                          0
16927  065710  000200                  AAPAT1:         200
16928  065712  000000                          0
16929  065714  000000                          0
16930  065716  000001                          1
16931  065720  000400                  AAPAT2:         400
16932  065722  000000                          0
16933  065724  000000                          0
16934  065726  000000                          0
16935  065730  000400                  AAPAT3:         400
16936  065732  000000                          0
16937  065734  000000                          0
16938  065736  000001                          1
16939  065740  000400                  AAPAT4:         400
16940  065742  000000                          0
16941  065744  100000                          100000
16942  065746  000000                          0
16943  065750  000200                  AAPAT5:         200
16944  065752  000001                          1
16945  065754  000000                          0
16946  065756  000000                          0
16947  065760  000400                  AAPAT6:         400
16948  065762  000001                          1
16949  065764  000000                          0
16950  065766  000000                          0
16951  065770                          AADONE:
  (1)  065770  004767  036556                  JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
16962                           ;*****************************************************************************
  (2)                           ;TEST 463       ADDF AND ADDD WITH E(AC) LESS THAN E(FSRC) TEST
  (3)                           ;*****************************************************************************
  (2)  065774                   TS463:
16963                           ;EXPONENT DIFFERENCE=57=71 (OCT) FD=1
16964  065774  012704  003200                  MOV     #3200,R4        ;SET FIV,FIV, AND FD
16965  066000  170104                          LDFPS   R4
16966  066002  012700  066450                  MOV     #CCP0,R0        ;SET AC0 OPERAND
16967  066006  172410                          LDD     (R0),AC0                ;AC0
16968  066010  012700  066470                  MOV     #CCP2,R0
16969  066014  172010                  CCX2:   ADDD    (R0),AC0        ;TEST INSTRUCTION
16970  066016  170205                          STFPS   R5              ;GET FPS
16971  066020  012700  066440                  MOV     #CCDAT0,R0      ;GET THE RESULT
```

```
16972  066024  174010                        STD     AC0,(R0)
16973  066026  012701  066470                MOV     #CCP2,R1        ;IS IT CORRECT
16974  066032  012702  000004                MOV     #4,R2
16975  066036  022021                CCX3:   CMP     (R0)+,(R1)+
16976  066040  001401                        BEQ     CCX6
  (2)  066042  104000                        EMT                     ;
16977  066044  077204                CCX6:   SOB     R2,CCX3
16978  066046  020405                        CMP     R4,R5           ;FPS CORRECT?
16979  066050  001401                        BEQ     CCX7
  (2)  066052  104000                        EMT                     ;
16980                                 ;EXPONENT DIFFERENCE=56-70 (OCT) FD=1
16981  066054  012704  003200        CCX7:   MOV     #3200,R4        ;SET FIV,FIV, AND FD
16982  066060  170104                        LDFPS   R4
16983  066062  012700  066450                MOV     #CCP0,R0        ;SET AC0 OPERAND
16984  066066  172410                        LDD     (R0),AC0
16985  066070  012700  066460                MOV     #CCP1,R0        ;FSRC
16986  066074  172010                CCX8:   ADDD    (R0),AC0        ;TEST INSTRUCTION
16987  066076  170205                        STFPS   R5              ;GET FPS
16988  066100  012700  066440                MOV     #CCDAT0,R0      ;GET THE RESULT
16989  066104  174010                        STD     AC0,(R0)
16990  066106  012701  066540                MOV     #CCP7,R1        ;IS IT CORRECT
16991  066112  012702  000004                MOV     #4,R2
16992  066116  022021                CCX9:   CMP     (R0)+,(R1)+
16993  066120  001401                        BEQ     CCX12
  (2)  066122  104000                        EMT                     ;
16994  066124  077204                CCX12:  SOB     R2,CCX9
16995  066126  020405                        CMP     R4,R5           ;FPS CORRECT?
16996  066130  001401                        BEQ     CCX13
  (2)  066132  104000                        EMT                     ;
16997                                 ;EXPONENT DIFFERENCE=25-31 (OCT) FD=0
16998  066134  012700  066450        CCX13:  MOV     #CCP0,R0        ;SET UP AC0 OPERAND.
16999  066140  172410                        LDD     (R0),AC0
17000  066142  012704  003000                MOV     #3000,R4        ;SET FIV,FIV. CLEAR FD.
17001  066146  170104                        LDFPS   R4
17002  066150  012700  066530                MOV     #CCP6,R0        ;FSRC
17003  066154  172010                CCX14:  ADDF    (R0),AC0        ;TEST INSTRUCTION
17004  066156  170205                        STFPS   R5
17005  066160  170011                        SETD                    ;REENTER DOUBLE MOVE
17006  066162  012700  066440                MOV     #CCDAT0,R0      ;GET THE RESULT
17007  066166  174010                        STD     AC0,(R0)
17008  066170  012701  066530                MOV     #CCP6,R1        ;IS THE RESULT CORRECT?
17009  066174  012702  000002                MOV     #2,R2
17010  066200  022021                CCX15:  CMP     (R0)+,(R1)+
17011  066202  001401                        BEQ     CCX18
  (2)  066204  104000                        EMT                     ;
17012  066206  077204                CCX18:  SOB     R2,CCX15
17013  066210  020405                        CMP     R4,R5
17014  066212  001401                        BEQ     CCX19
  (2)  066214  104000                        EMT                     ;
17015                                 ;EXPONENT DIFFERENCE=24-30 (OCT) FD=0
17016  066216  012700  066500        CCX19:  MOV     #CCP3,R0        ;SET UP AC0 OPERAND.
17017  066222  172410                        LDD     (R0),AC0
17018  066224  012704  003000                MOV     #3000,R4        ;SET FIV,FIV. CLEAR FD.
17019  066230  170104                        LDFPS   R4
17020  066232  012700  066520                MOV     #CCP5,R0        ;FSRC
17021  066236  172010                CCX20:  ADDF    (R0),AC0        ;TEST INSTRUCTION
```

```
17022  066240  170205                  STFPS   R5
17023  066242  170011                  SETD                        ;REENTER DOUBLE MOVE
17024  066244  012700  066440          MOV     #CCDATO,R0          ;GET THE RESULT
17025  066250  174010                  STD     AC0,(R0)
17026  066252  012701  066550          MOV     #CCP10,R1           ;IS THE RESLT CORRECT?
17027  066256  012702  000002          MOV     #2,R2
17028  066262  022021          CCX21:  CMP     (R0)+,(R1)+
17029  066264  001401                  BEQ     CCX24
  (2)  066266  104000                  EMT                         ;
17030  066270  077204          CCX24:  SOB     R2,CCX21
17031  066272  020405                  CMP     R4,R5
17032  066274  001401                  BEQ     CCX25
  (2)  066276  104000                  EMT                         ;
17033                           ;EXPONENT DIFFERENCE=1 FD=1
17034  066300  012704  003200  CCX25:  MOV     #3200,R4            ;SET FIV,FIV, AND FD
17035  066304  170104                  LDFPS   R4
17036  066306  012700  066450          MOV     #CCP0,R0            ;SET AC0 OPERAND
17037  066312  172410                  LDD     (R0),AC0
17038  066314  012700  066500          MOV     #CCP3,R0            ;FSRC
17039  066320  172010          CCX26:  ADDD    (R0),AC0            ;TEST INSTRUCTION
17040  066322  170205                  STFPS   R5                  ;GET FPS
17041  066324  012700  066440          MOV     #CCDATO,R0          ;GET THE RESULT
17042  066330  174010                  STD     AC0,(R0)
17043  066332  012701  066560          MOV     #CCP11,R1           ;IS IT CORRECT
17044  066336  012702  000004          MOV     #4,R2
17045  066342  022021          CCX27:  CMP     (R0)+,(R1)+
17046  066344  001401                  BEQ     CCX30
  (2)  066346  104000                  EMT                         ;
17047  066350  077204          CCX30:  SOB     R2,CCX27
17048  066352  020405                  CMP     R4,R5               ;FPS CORRECT?
17049  066354  001401                  BEQ     CCX31
  (2)  066356  104000                  EMT                         ;
17050                           ;EXPONENT DIFFERENCE=100=144 (OCT) FD=1
17051  066360  012704  003200  CCX31:  MOV     #3200,R4            ;SET FIV,FIV, AND FD
17052  066364  170104                  LDFPS   R4
17053  066366  012700  066450          MOV     #CCP0,R0            ;SET AC0 OPERAND
17054  066372  172410                  LDD     (R0),AC0
17055  066374  012700  066510          MOV     #CCP4,R0            ;FSRC
17056  066400  172010          CCX32:  ADDD    (R0),AC0            ;TEST INSTRUCTION
17057  066402  170205                  STFPS   R5                  ;GET FPS
17058  066404  012700  066440          MOV     #CCDATO,R0          ;GET THE RESULT
17059  066410  174010                  STD     AC0,(R0)
17060  066412  012701  066510          MOV     #CCP4,R1            ;IS IT CORRECT
17061  066416  012702  000004          MOV     #4,R2
17062  066422  022021          CCX33:  CMP     (R0)+,(R1)+
17063  066424  001401                  BEQ     CCX36
  (2)  066426  104000                  EMT                         ;
17064  066430  077204          CCX36:  SOB     R2,CCX33
17065  066432  020405                  CMP     R4,R5               ;FPS CORRECT?
17066  066434  001461                  BEQ     CCXDONE
  (2)  066436  104000                  EMT                         ;
17067  066440  000000          CCDATO: 0
17068  066442  000000                  0
17069  066444  000000                  0
17070  066446  000000                  0
17071  066450  000200          CCP0:   200                         ;E(AC)=1
```

```
17072  066452  000000                          0
17073  066454  000000                          0
17074  066456  000000                          0
17075  066460  016200                CCP1:     16200                ;E(FSRC)=E(AC)+56=57
17076  066462  000000                          0                    ;                =71(OCT)
17077  066464  000000                          0
17078  066466  000000                          0
17079  066470  016400                CCP2:     16400                ;E(FSRC)=E(AC)+57=58
17080  066472  000000                          0                    ;                =72(OCT)
17081  066474  000000                          0
17082  066476  000000                          0
17083  066500  000400                CCP3:     400                  ;E(FSRC)=E(AC)+1=2
17084  066502  000000                          0
17085  066504  000000                          0
17086  066506  000000                          0
17087  066510  031200                CCP4:     31200                ;E(FSRC)=E(AC)+100=101=145(OCT)
17088  066512  000000                          0
17089  066514  000000                          0
17090  066516  000000                          0
17091  066520  006200                CCP5:     6200                 ;E(FSRC)=E(AC)+24=25=31(OCT)
17092  066522  000000                          0
17093  066524  000000                          0
17094  066526  000000                          0
17095  066530  006400                CCP6:     6400                 ;E(FSRC)=E(AC)+25=26=32(OCT)
17096  066532  000000                          0
17097  066534  000000                          0
17098  066536  000000                          0
17099  066540  016200                CCP7:     16200                ;CCP1 RES
17100  066542  000000                          0
17101  066544  000000                          0
17102  066546  000001                          1
17103  066550  006200                CCP10:    6200                 ;CCP5 RES
17104  066552  000001                          1
17105  066554  000000                          0
17106  066556  000000                          0
17107  066560  000500                CCP11:    500                  ;CCP3 RES
17108  066562  000000                          0
17109  066564  000000                          0
17110  066566  000000                          0
17111  066570  000200                CCP12:    200                  ;BAD CONSTANT
17112  066572  000000                          0                    ;RES CCP2,CCP4
17113  066574  000000                          0
17114  066576  000000                          0
17115
17116  066600                          CCXDONE:
  (1)  066600  004767  035746                  JSR     PC,..RSET    ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                               ;SEE IF THE USER HAS EXPRESSED
  (1)                                                               ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                               ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                               ;THE USER TYPED CONTROL G?).
17117
17128                             ;*****************************************************************************
  (2)                             ;TEST 464       ADDF AND ADDD WITH E(AC) GREATER THAN E(FSRC) TEST
  (3)                             ;*****************************************************************************
  (2)  066604                     TS464:
17129                             ;EXPONENT DIFFERENCE=57=71 (OCT) FD=1
```

```
17130  066604  012704  003200            MOV     #3200,R4        ;SET FIV FIV, AND FD
17131  066610  170104                    LDFPS   R4
17132  066612  012700  067300            MOV     #BBPAT2,R0      ;SET AC0 OPERAND.
17133  066616  172410                    LDD     (R0),AC0
17134  066620  012700  067270            MOV     #BBPAT1,R0      ;FSRC
17135  066624  172010            BB2:    ADDD    (R0),AC0        ;TEST INSTRUCTION
17136  066626  170205                    STFPS   R5
17137  066630  012700  067250    BB3:    MOV     #BBDAT0,R0      ;GET THE RESULT
17138  066634  174010                    STD     AC0,(R0)
17139  066636  012701  067300            MOV     #BBPAT2,R1      ;RESULT CORRECT?
17140  066642  012702  000004            MOV     #4,R2
17141  066646  022021            BB4:    CMP     (R0)+,(R1)+
17142  066650  001401                    BEQ     BB5
  (2)  066652  104000                    EMT                     ;
17143  066654  077204            BB5:    SOB     R2,BB4
17144                                                            ;WAS FPS CORRECT?
17145  066656  020405                    CMP     R4,R5
17146  066660  001401                    BEQ     BB6
  (2)  066662  104000                    EMT                     ;
17147                            ;EXPONENT DIFFERENCE=56-70 (OCT) FD=1
17148  066664  012704  003200    BB6:    MOV     #3200,R4        ;SET FIV,FIV, AND FD
17149  066670  170104                    LDFPS   R4
17150  066672  012700  067320            MOV     #BBPAT4,R0      ;SET AC0 OPERAND
17151  066676  172410                    LDD     (R0),AC0
17152  066700  012700  067270            MOV     #BBPAT1,R0      ;FSRC
17153  066704  172010            BB7:    ADDD    (R0),AC0        ;TEST INSTRUCTION
17154  066706  170205                    STFPS   R5              ;GET FPS
17155  066710  012700  067250            MOV     #BBDAT0,R0      ;GET THE RESULT
17156  066714  174010                    STD     AC0,(R0)
17157  066716  012701  067360            MOV     #BBP10,R1       ;IS IT CORRECT
17158  066722  012702  000004            MOV     #4,R2
17159  066726  022021            BB10:   CMP     (R0)+,(R1)+
17160  066730  001401                    BEQ     BB13
  (2)  066732  104000                    EMT                     ;
17161  066734  077204            BB13:   SOB     R2,BB10
17162  066736  020405                    CMP     R4,R5           ;FPS CORRECT?
17163  066740  001401                    BEQ     BB14
  (2)  066742  104000                    EMT                     ;
17164                            ;EXPONENT DIFFERENCE=25-31 (OCT) FD=0
17165  066744  012700  067260    BB14:   MOV     #BBPAT0,R0      ;SET UP AC0 OPERAND
17166  066750  172410                    LDD     (R0),AC0
17167  066752  012704  003000            MOV     #3000,R4        ;SET FIV AND FIV
17168                                                            ;CLEAR FD
17169  066756  170104                    LDFPS   R4
17170  066760  012700  067270            MOV     #BBPAT1,R0      ;FSRC
17171  066764  172010            BB15:   ADDF    (R0),AC0        ;TEST INSTRUCTION
17172  066766  170205                    STFPS   R5
17173  066770  170011                    SETD                    ;REENTED DOUBLE MODE.
17174  066772  012700  067250            MOV     #BBDAT0,R0      ;GET THE RESULT
17175  066776  174010                    STD     AC0,(R0)
17176  067000  012701  067260            MOV     #BBPAT0,R1      ;IS THE RESULT
17177  067004  012702  000002            MOV     #2,R2           ;CORRECT?
17178  067010  022021            BB16:   CMP     (R0)+,(R1)+
17179  067012  001401                    BEQ     BB17
  (2)  067014  104000                    EMT                     ;
17180  067016  077204            BB17:   SOB     R2,BB16
```

```
17181  067020  020405                    CMP     R4,R5           ;IS FPS CORRECT?
17182  067022  001401                    BEQ     BB20
  (2)  067024  104000                    EMT                     ;
17183                            ;EXPONENT DIFFERENCE=24=30 (OCT)
17184  067026  012700  067310    BB20:   MOV     #BBPAT3,R0      ;SET UP AC0 OPERAND.
17185  067032  172410                    LDD     (R0),AC0
17186  067034  012704  003000            MOV     #3000,R4        ;SET FIU,FIV. CLEAR FD.
17187  067040  170104                    LDFPS   R4
17188  067042  012700  067270            MOV     #BBPAT1,R0      ;FSRC
17189  067046  172010            BB21:   ADDF    (R0),AC0        ;TEST INSTRUCTION
17190  067050  170205                    STFPS   R5
17191  067052  170011                    SETD                    ;REENTER DOUBLE MODE
17192  067054  012700  067250            MOV     #BBDAT0,R0      ;GET THE RESULT
17193  067060  174010                    STD     AC0,(R0)
17194  067062  012701  067350            MOV     #BBP7,R1        ;IS THE RESULT CORRECT?
17195  067066  012702  000002            MOV     #2,R2
17196  067072  022021            BB22:   CMP     (R0)+,(R1)+
17197  067074  001401                    BEQ     BB25
  (2)  067076  104000                    EMT                     ;
17198  067100  077204            BB25:   SOB     R2,BB22
17199  067102  020405                    CMP     R4,R5
17200  067104  001401                    BEQ     BB26
  (2)  067106  104000                    EMT                     ;
17201                            ;EXPONENT DIFFERENCE=1
17202  067110  012704  003200    BB26:   MOV     #3200,R4
17203  067114  170104                    LDFPS   R4              ;SET UP AC0 OPERAND
17204  067116  012700  067330            MOV     #BBPAT5,R0
17205  067122  172410                    LDD     (R0),AC0
17206  067124  012700  067270            MOV     #BBPAT1,R0      ;FSRC
17207  067130  172010            BB27:   ADDD    (R0),AC0        ;TEST INSTRUCTION
17208  067132  170205                    STFPS   R5
17209  067134  012700  067250            MOV     #BBDAT0,R0      ;GET THE RESULT.
17210  067140  174010                    STD     AC0,(R0)
17211  067142  012701  067370            MOV     #BBP11,R1       ;IS IT CORRECT?
17212  067146  012702  000004            MOV     #4,R2
17213  067152  022021            BB30:   CMP     (R0)+,(R1)+
17214  067154  001401                    BEQ     BB31
  (2)  067156  104000                    EMT                     ;
17215  067160  077204            BB31:   SOB     R2,BB30
17216  067162  020405                    CMP     R4,R5           ;IS FPS CORRECT
17217  067164  001401                    BEQ     BB32
  (2)  067166  104000                    EMT                     ;
17218                            ;EXPONENT DIFFERENCE=100=144 (OCT)
17219  067170  012704  003200    BB32:   MOV     #3200,R4
17220  067174  170104                    LDFPS   R4              ;SET FIV,FIV AND FD
17221  067176  012700  067340            MOV     #BBPAT6,R0      ;SET UP AC0 OPERAND.
17222  067202  172410                    LDD     (R0),AC0
17223  067204  012700  067270            MOV     #BBPAT1,R0      ;FSRC
17224  067210  172010            BB33:   ADDD    (R0),AC0        ;TEST INSTRUCTION
17225  067212  170205                    STFPS   R5
17226  067214  012700  067250            MOV     #BBDAT0,R0      ;GET THE RESULT
17227  067220  174010                    STD     AC0,(R0)
17228  067222  012701  067340            MOV     #BBPAT6,R1      ;IS IT CORRECT
17229  067226  012702  000004            MOV     #4,R2
17230  067232  022021            BB34:   CMP     (R0)+,(R1)+
17231  067234  001401                    BEQ     BB35
```

```
  (2)  067236  104000                          EMT                     ;
17232  067240  077204                  BB35:   SOB     R2,BB34
17233  067242  020405                          CMP     R4,R5           ;IS FPS CORRECT
17234  067244  001455                          BEQ     BBDONE
  (2)  067246  104000                          EMT                     ;
17235  067250  000000                  BBDAT0: 0
17236  067252  000000                          0
17237  067254  000000                          0
17238  067256  000000                          0
17239  067260  006400                  BBPAT0: 6400                    ;F(AC)=E(FSRC)+25=26
17240  067262  000000                          0                       ;                  =32(OCT)
17241  067264  000000                          0
17242  067266  000000                          0
17243  067270  000200                  BBPAT1: 200
17244  067272  000000                          0                       ;E(FSRC)=1
17245  067274  000000                          0
17246  067276  000000                          0
17247  067300  016400                  BBPAT2: 16400
17248  067302  000000                          0                       ;E(AC)=E(FSRC)+57=58
17249  067304  000000                          0                       ;                  =72(OCT)
17250  067306  000000                          0
17251  067310  006200                  BBPAT3: 6200                    ;E(AC)=E(FSRC)+24=25
17252  067312  000000                          0                       ;                  =31(OCT)
17253  067314  000000                          0
17254  067316  000000                          0
17255  067320  016200                  BBPAT4: 16200                   ;E(AC)=E(FSR6)+56=57
17256  067322  000000                          0                       ;                  =71(OCT)
17257  067324  000000                          0
17258  067326  000000                          0
17259  067330  000400                  BBPAT5: 400                     ;E(AC)=E(FSRC)+1=2
17260  067332  000000                          0
17261  067334  000000                          0
17262  067336  000000                          0
17263  067340  031200                  BBPAT6: 31200                   ;E(AC)=E(FSRC)+100=101
17264  067342  000000                          0                       ;                  =145(OCT)
17265  067344  000000                          0
17266  067346  000000                          0
17267  067350  006200                  BBP7:   6200                    ;BBPAT3 RES
17268  067352  000001                          1
17269  067354  000000                          0
17270  067356  000000                          0
17271  067360  016200                  BBP10:  16200                   ;BBPAT4 RES
17272  067362  000000                          0
17273  067364  000000                          0
17274  067366  000001                          1
17275  067370  000500                  BBP11:  500                     ;BBPAT5 RES
17276  067372  000000                          0
17277  067374  000000                          0
17278  067376  000000                          0
17279  067400                          BBDONE:
  (1)  067400  004767  035146                  JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
17287                                  ;*****************************************************************************
```

```
  (2)                                   ;TEST 465       ADDD WITH NEGATVE OPRANDS TEST
  (3)                                   ;*****************************************************************************************
  (2)  067404                           TS465:
17288                                   ;BOTH OPERANDS NEGATIVE
17289  067404  012704  003200                 MOV     #3200,R4        ;SET FIO, FIV, AND FD
17290  067410  170104                         LDFPS   R4
17291  067412  012700  070170                 MOV     #DDP1,R0                ;SET ACO OPERAND
17292  067416  172410                         LDD     (R0),AC0
17293  067420  012700  070170                 MOV     #DDP1,R0                ;ESRC
17294  067424  172010                 DD2:    ADDD    (R0),AC0                ;TEST INSTRUCTION
17295  067426  170205                         STFPS   R5                      ;GET FPS
17296  067430  012700  070150                 MOV     #DDDAT0,R0              ;GET THE RESULT
17297  067434  174010                         STD     AC0,(R0)
17298  067436  012701  070270                 MOV     #DDP9,R1                ;IS IT CORRECT
17299  067442  012702  000004                 MOV     #4,R2
17300  067446  022021                 DD3:    CMP     (R0)+,(R1)+
17301  067450  001401                         BEQ     DD6
  (2)  067452  104000                         EMT                     ;
17302  067454  077204                 DD6:    SOB     R2,DD3
17303  067456  052704  000010                 BIS     #10,R4
17304  067462  020405                         CMP     R4,R5                   ;FPS CORRECT?
17305  067464  001401                         BEQ     DD7
  (2)  067466  104000                         EMT                     ;
17306                                   ;AC POS FSRC NEG        AC=-FSRC
17307  067470  012704  003200         DD7:    MOV     #3200,R4                ;SET FIO, FIV, AND FD
17308  067474  170104                         LDFPS   R4
17309  067476  012700  070200                 MOV     #DDP2,R0                ;SET ACO OPERAND
17310  067502  172410                         LDD     (R0),AC0
17311  067504  012700  070170                 MOV     #DDP1,R0                ;FSPC
17312  067510  172010                 DD8:    ADDD    (R0),AC0                ;TEST INSTRUCTION
17313  067512  170205                         STFPS   R5                      ;GET FPS
17314  067514  012700  070150                 MOV     #DDDAT0,R0              ;GET THE RESULT
17315  067520  174010                         STD     AC0,(R0)
17316  067522  012701  070160                 MOV     #DDP0,R1                ;IS IT CORRECT
17317  067526  012702  000004                 MOV     #4,R2
17318  067532  022021                 DD10:   CMP     (R0)+,(R1)+
17319  067534  001401                         BEQ     DD11
  (2)  067536  104000                         EMT                     ;
17320  067540  077204                 DD11:   SOB     R2,DD10
17321  067542  052704  000004                 BIS     #4,R4
17322  067546  020405                         CMP     R4,R5                   ;FPS CORRECT?
17323  067550  001401                         BEQ     DD12
  (2)  067552  104000                         EMT                     ;
17324                                   ;AC NEG FSRC POS                AC=-FSRC
17325  067554  012704  003200         DD12:   MOV     #3200,R4                ;SET FIU, FIV, AND FD
17326  067560  170104                         LDFPS   R4
17327  067562  012700  070170                 MOV     #DDP1,R0                ;SET ACO OPERAND
17328  067566  172410                         LDD     (R0),AC0
17329  067570  012700  070200                 MOV     #DDP2,R0                ;FSRC
17330  067574  172010                 DD13:   ADDD    (R0),AC0                ;TEST INSTRUCTION
17331  067576  170205                         STFPS   R5                      ;GET FPS
17332  067600  012700  070150                 MOV     #DDDAT0,R0              ;GET THE RESULT
17333  067604  174010                         STD     AC0,(R0)
17334  067606  012701  070160                 MOV     #DDP0,R1                ;IS IT CORRECT
17335  067612  012702  000004                 MOV     #4,R2
17336  067616  022021                 DD14:   CMP     (R0)+,(R1)+
```

```
17337  067620  001401                      BEQ     DD15
  (2)  067622  104000                      EMT                             ;
17338  067624  077204              DD15:   SOB     R2,DD14
17339  067626  052704  000004              BIS     #4,R4
17340  067632  020405                      CMP     R4,R5                   ;EPS CORRECT?
17341  067634  001401                      BEQ     DD16
  (2)  067636  104000                      EMT                             ;
17342                          ;AC0 POC     FSRC    NEG     /AC/ > /FSRC/
17343  067640  012704  003200      DD16:   MOV     #3200,R4                ;SET FIV, FIV AND FD
17344  067644  170104                      LDFPS   R4
17345  067646  012700  070210              MOV     #DDP3,R0                ;SET AC0 OPERAND
17346  067652  172410                      LDD     (R0),AC0
17347  067654  012700  070240              MOV     #DDP6,R0                ;ESPC
17348  067660  172010              DD17:   ADDD    (R0),AC0                ;TEST INSTRUCTION
17349  067662  170205                      STFPS   R5                      ;GET FPS
17350  067664  012700  070150              MOV     #DDDAT0,R0              ;GET THE RESULT
17351  067670  174010                      STD     AC0,(R0)
17352  067672  012701  070250              MOV     #DDP7,R1                ;IS IT CORRECT
17353  067676  012702  000004              MOV     #4,R2
17354  067702  022021              DD18:   CMP     (R0)+,(R1)+
17355  067704  001401                      BEQ     DD21
  (2)  067706  104000                      EMT                             ;
17356  067710  077204              DD21:   SOB     R2,DD18
17357  067712  020405                      CMP     R4,R5                   ;EPS CORRECT?
17358  067714  001401                      BEQ     DD22
  (2)  067716  104000                      EMT                             ;
17359                          ;AC NEG FSRC    POS     /FSRC/ > /AC/
17360  067720  012704  003200      DD22:   MOV     #3200,R4                ;SET FIO,FIV, AND FD
17361  067724  170104                      LDFPS   R4
17362  067726  012700  070240              MOV     #DDP6,R0                ;SET AC0 OPERAND
17363  067732  172410                      LDD     (R0),AC0
17364  067734  012700  070210              MOV     #DDP3,R0                ;FSPC
17365  067740  172010              DD23:   ADDD    (R0),AC0                ;TEST INSTRUCTION
17366  067742  170205                      STFPS   R5                      ;GET FPS
17367  067744  012700  070150              MOV     #DDDAT0,R0              ;GET THE RESULT
17368  067750  174010                      STD     AC0,(R0)
17369  067752  012701  070250              MOV     #DDP7,R1                ;IS IT CORRECT?
17370  067756  012702  000004              MOV     #4,R2
17371  067762  022021              DD24:   CMP     (R0)+,(R1)+
17372  067764  001401                      BEQ     DD27
  (2)  067766  104000                      EMT                             ;
17373  067770  077204              DD27:   SOB     R2,DD24
17374  067772  020405                      CMP     R4,R5                   ;FPS CORRECT?
17375  067774  001401                      BEQ     DD30
  (2)  067776  104000                      EMT                             ;
17376                          ;AC0 POS     FSRC    NEG     /AC/</FRSRC/
17377  070000  012704  003200      DD30:   MOV     #3200,R4                ;SET FIO,FIV,AND FD
17378  070004  170104                      LDFPS   R4
17379  070006  012700  070220              MOV     #DDP4,R0                ;SET AC0 OPERAND
17380  070012  172410                      LDD     (R0),AC0
17381  070014  012700  070230              MOV     #DDP5,R0                ;FSPC
17382  070020  172010              DD31:   ADDD    (R0),AC0                ;TEST INSTRUCTION
17383  070022  170205                      STFPS   R5                      ;GET FPS
17384  070024  012700  070150              MOV     #DDDAT0,R0              ;GET THE RESULT
17385  070030  174010                      STD     AC0,(R0)
17386  070032  012701  070260              MOV     #DDP8,R1                ;IS IT CORRECT
```

```
17387  070036  012702  000004            MOV     #4,R2
17388  070042  022021                DD32: CMP     (R0)+,(R1)+
17389  070044  001401                      BEQ     DD35
  (2)  070046  104000                      EMT                             ;
17390  070050  077204                DD35: SOB     R2,DD32
17391  070052  052704  000010              BIS     #10,R4
17392  070056  020405                      CMP     R4,R5                   ;FPS CORRECT?
17393  070060  001401                      BEQ     DD36
  (2)  070062  104000                      EMT                             ;
17394                                ;ACO NEG      FSRC    POS             /FSRC/</AC/
17395  070064  012704  003200        DD36: MOV     #3200,R4                ;SET FIO, FIV, AND FD
17396  070070  170104                      LDFPS   R4
17397  070072  012700  070230              MOV     #DDP5,R0                ;SET ACO OPERAND
17398  070076  172410                      LDD     (R0),AC0
17399  070100  012700  070220              MOV     #DDP4,R0                ;FSPC
17400  070104  172010                DD37: ADDD    (R0),AC0                ;TEST INSTRUCTION
17401  070106  170205                      STFPS   R5                      ;GET FPS
17402  070110  012700  070150              MOV     #DDDAT0,R0              ;GET THE RESULT
17403  070114  174010                      STD     AC0,(R0)
17404  070116  012701  070260              MOV     #DDP8,R1                ;IS IT CORRECT
17405  070122  012702  000004              MOV     #4,R2
17406  070126  022021                DD38: CMP     (R0)+,(R1)+
17407  070130  001401                      BEQ     DD41
  (2)  070132  104000                      EMT                             ;
17408  070134  077204                DD41: SOB     R2,DD38
17409  070136  052704  000010              BIS     #10,R4
17410  070142  020405                      CMP     R4,R5                   ;FPS CORRECT?
17411  070144  001455                      BEQ     DDDONE
  (2)  070146  104000                      EMT                             ;
17412  070150  000000              DDDAT0: 0
17413  070152  000000                      0
17414  070154  000000                      0
17415  070156  000000                      0
17416  070160  000000              DDP0:   0
17417  070162  000000                      0
17418  070164  000000                      0
17419  070166  000000                      0
17420  070170  100200              DDP1:   100200                          ;-DDP2
17421  070172  000000                      0
17422  070174  000000                      0
17423  070176  000000                      0
17424  070200  000200              DDP2:   200                             ;-DDP1
17425  070202  000000                      0
17426  070204  000000                      0
17427  070206  000000                      0
17428  070210  001100              DDP3:   1100                            ;EXP=4
17429  070212  000000                      0                               ;FRAC=...110...
17430  070214  000000                      0
17431  070216  000000                      0
17432  070220  000600              DDP4:   600                             ;EXP=3
17433  070222  000000                      0                               ;FRAC=...100...
17434  070224  000000                      0
17435  070226  000000                      0
17436  070230  101100              DDP5:   101100                          ;-DDP3
17437  070232  000000                      0
17438  070234  000000                      0
```

```
070236  000000                          0
070240  100600                  DDP6:   100600                          ;-DDP4
070242  000000                          0
070244  000000                          0
070246  000000                          0
070250  001000                  DDP7:   1000                            ;DDP3+DDP6
070252  000000                          0
070254  000000                          0
070256  000000                          0
070260  101000                  DDP8:   101000                          ;DDP5+DDP4
070262  000000                          0
070264  000000                          0
070266  000000                          0
070270  100400                  DDP9:   100400                          ;DDP1+DDP1
070272  000000                          0
070274  000000                          0
070276  000000                          0
070300                          DDDONE:
070300  004767  034246                  JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
                                                                ;SEE IF THE USER HAS EXPRESSED
                                                                ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                ;THE USER TYPED CONTROL G?).
                                ;*******************************************************************************
                                ;TEST 466       SUBD TEST
                                ;*******************************************************************************
070304                          TS466:
                                ;       USE POSITIVE OPERANDS
070304  012704  003200                  MOV     #3200,R4                ;SET FIU, FIV, AND FD
070310  170104                          LDFPS   R4
070312  012700  070476                  MOV     #EEP1,R0                ;SET AC0 OPERAND
070316  172410                          LDD     (R0),AC0
070320  012700  070476                  MOV     #EEP1,R0                ;FSPC
070324  173010                  EE2:    SUBD    (R0),AC0                ;TEST INSTRUCTION
070326  170205                          STFPS   R5                      ;GET FPS
070330  012700  070454                  MOV     #EEDATO,R0              ;GET THE RESULT
070334  174010                          STD     AC0,(R0)
070336  012701  070464                  MOV     #EEP0,R1                ;IS IT CORRECT?
070342  012702  000004                  MOV     #4,R2
070346  022021                  EE3:    CMP     (R0)+,(R1)+
070350  001401                          BEQ     EE6
070352  104000                          EMT                     ;
070354  077204                  EE6:    SOB     R2,EE3
070356  052704  000004                  BIS     #4,R4
070362  020405                          CMP     R4,R5                   ;FPS CORRECT?
070364  001401                          BEQ     EE7
070366  104000                          EMT                     ;
                                        ;USE NEGATIVE OPERANDS
070370  012704  003200          EE7:    MOV     #3200,R4                ;SET FIO, FIV, AND FD
070374  170104                          LDFPS   R4
070376  012700  070516                  MOV     #EEP3,R0                ;SET AC0 OPERAND
070402  172410                          LDD     (R0),AC0
070404  012700  070516                  MOV     #EEP3,R0                ;FSPC
070410  173010                  EE8:    SUBD    (R0),AC0                ;TEST INSTRUCTION
070412  170205                          STFPS   R5                      ;GET FPS
070414  012700  070454                  MOV     #EEDATO,R0              ;GET THE RESULT
```

```
17492  070420  174010                    STD     AC0,(R0)
17493  070422  012701  070464            MOV     @EEP0,R1            ;IS IT CORRECT?
17494  070426  012702  000004            MOV     #4,R2
17495  070432  022021            EE9:    CMP     (R0)+,(R1)+
17496  070434  001401                    BEQ     EE12
  (2)  070436  104000                    EMT                         ;
17497  070440  077204            EE12:   SOB     R2,EE9
17498  070442  052704  000004            BIS     #4,R4
17499  070446  020405                    CMP     R4,R5               ;FPS CORRECT?
17500  070450  001432                    BEQ     EEDONE
  (2)  070452  104000                    EMT                         ;
17501  070454  000000            EEDAT0: 0
17502  070456  000000                    0
17503  070460  000000                    0
17504  070462  000000                    0
17505  070464  000000            EEP0:   0
17506  070466  000000                    0
17507  070470  000000            00000
17508  070472  000000                    0
17509  070474  000000                    0
17510  070476  000200            EEP1:   200
17511  070500  000000                    0
17512  070502  000000                    0
17513  070504  000000                    0
17514  070506  000400            EEP2:   400
17515  070510  000000                    0
17516  070512  000000                    0
17517  070514  000000                    0
17518  070516  100200            EEP3:   100200
17519  070520  000000                    0
17520  070522  000000                    0
17521  070524  000000                    0
17522  070526  100400            EEP4:   100400
17523  070530  000000                    0
17524  070532  000000                    0
17525  070534  000000                    0
17526  070536                    EEDONE:
  (1)  070536  004767  034010            JSR     PC,.RSET            ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                ;THE USER TYPED CONTROL G?).
17536                       ;*****************************************************************************
  (2)                       ;TEST 467       NORMALIZE ALGORITHM TEST
  (3)                       ;*****************************************************************************
  (2)  070542               TS467:
17537                       ;USE DATA PATTERNS THAT REQUIRE ONLY ONE LEFT SHIFT TO NORMALIZE
17538  070542  012704  003200            MOV     #3200,R4            ;SET FIO, FIV, AND FD
17539  070546  170104                    LDFPS   R4
17540  070550  012700  070732            MOV     #FFP2,R0            ;SET AC0 OPERAND
17541  070554  172410                    LDD     (R0),AC0
17542  070556  012700  070742            MOV     #FFP3,R0            ;FSPC
17543  070562  172010            FF2:    ADDD    (R0),AC0            ;TEST INSTRUCTION
17544  070564  170205                    STFPS   R5                  ;GET FPS
17545  070566  012700  070702            MOV     #FFDAT0,R0          ;GET THE RESULT
17546  070572  174010                    STD     AC0,(R0)
```

```
 17547  070574  012701  070752                 MOV     #FFP4, R1               ;IS IT CORRECT
```

```
17549  070600  012702  000004          MOV     #4,R2
17550  070604  022021          FF3:    CMP     (R0)+,(R1)+
17551  070606  001401                  BEQ     FF4
  (2)  070610  104000                  EMT                     ;
17552  070612  077204          FF4:    SOB     R2,FF3
17553  070614  020405                  CMP     R4,R5                   ;FPS CORRECT?
17554  070616  001401                  BEQ     FF5
  (2)  070620  104000                  EMT                     ;
17555                          ;USE DATA PATTERNS WHICH REQUIRE 56 LEFT SHIFTS TO NORMALIZE
17556                          ;THE RESULT
17557  070622  012704  003200  FF5:    MOV     #3200,R4                ;SET FIU, FIV, AND FD
17558  070626  170104                  LDFPS   R4
17559  070630  012700  070712          MOV     #FFP0,R0                ;SET AC0 OPERAND
17560  070634  172410                  LDD     (R0),AC0
17561  070636  012700  070722          MOV     #FFP1,R0                ;FSRC
17562  070642  172010          FF6:    ADDD    (R0),AC0                ;TEST INSTRUCTION
17563  070644  170205                  STFPS   R5                      ;GET FPS
17564  070646  012700  070702          MOV     #FFDAT0,R0              ;GET THE RESULT
17565  070652  174010                  STD     AC0,(R0)
17566  070654  012701  070752          MOV     #FFP4,R1                ;IS IT CORRECT
17567  070660  012702  000004          MOV     #4,R2
17568  070664  022021          FF7:    CMP     (R0)+,(R1)+
17569  070666  001401                  BEQ     FF10
  (2)  070670  104000                  EMT                     ;
17570  070672  077204          FF10:   SOB     R2,FF7
17571  070674  020405                  CMP     R4,R5                   ;FPS CORRECT?
17572  070676  001431                  BEQ     FFDONE
  (2)  070700  104000                  EMT                     ;
17573
17574
17575  070702  000000          FFDAT0: 0
17576  070704  000000                  0
17577  070706  000000                  0
17578  070710  000000                  0
17579
17580  070712  016000          FFP0:   16000
17581  070714  000000                  0
17582  070716  000000                  0
17583  070720  000001                  1
17584  070722  116000          FFP1:   116000
17585  070724  000000                  0
17586  070726  000000                  0
17587  070730  000000                  0
17588  070732  000500          FFP2:   500
17589  070734  000000                  0
17590  070736  000000                  0
17591  070740  000000                  0
17592  070742  100400          FFP3:   100400
17593  070744  000000                  0
17594  070746  000000                  0
17595  070750  000000                  0
17596  070752  000200          FFP4:   200                     ;FFP4=FFP0+FFP1
17597  070754  000000                  0                       ;     =FFP3+FFP4
17598  070756  000000                  0
17599  070760  000000                  0
17600
```

```
17601  070762                          FFDONE:
  (1)  070762  004767  033564                JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                 ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                 ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                 ;THE USER TYPED CONTROL G?).
17602
17603
17604
17605                                 ;*************************************************************************
17606                                 ;*************************************************************************
17607                                 ;FLOATING POINT SECOND PART
17608                                 ;*************************************************************************
17609                                 ;*************************************************************************
17610
17611
17622                                 ;*************************************************************************************
  (2)                                 ;TEST 470       ROUND\TRUNK TEST
  (3)                                 ;*************************************************************************************
  (2)  070766                         TS470:
17623
17624                                         ;ROUND AND NORMALIZE TEST
17625
17626  070766  012704  003200                MOV     #3200,R4                ;SET FIU, FIV, AND FD
17627  070772  170104                        LDFPS   R4
17628  070774  012700  071376                MOV     #MMP0,R0                ;SET AC0 OPERAND
17629  071000  172410                        LDD     (R0),AC0
17630  071002  012700  071406                MOV     #MMP1,R0                ;FSPC
17631  071006  172010                 MM2:   ADDD    (R0),AC0                ;TEST INSTRUCTION
17632  071010  170205                        STFPS   R5                      ;GET FPS
17633  071012  012700  071366                MOV     #MMDAT0,R0              ;GET THE RESULT
17634  071016  174010                        STD     AC0,(R0)
17635  071020  012701  071416                MOV     #MMP2,R1                ;IS IT CORRECT
17636  071024  012702  000004                MOV     #4,R2
17637  071030  022021                 MM3:   CMP     (R0)+,(R1)+
17638  071032  001401                        BEQ     MM6
  (2)  071034  104000                        EMT                     ;
17639  071036  077204                 MM6:   SOB     R2,MM3
17640  071040  020405                        CMP     R4,R5           ;FPS CORRECT?
17641  071042  001401                        BEQ     MM7
  (2)  071044  104000                        EMT                     ;
17642
17643                                 ;THIS IS A TEST OF THE ABILITY
17644                                 ;OF NORMALIZE TO PRODUCE A ZERO EXP. AND
17645                                 ;OF THE R\T ALGORITHM TO PROPERLY SET THE FPS
17646
17647  071046  012704  043200        MM7:   MOV     #043200,R4              ;SET FIU,FIV,AND FD
17648                                                                        ;FID
17649  071052  170104                        LDFPS   R4
17650  071054  012700  071446                MOV     #MMP5,R0                ;SET AC0 OPERAND
17651  071060  172410                        LDD     (R0),AC0
17652  071062  012700  071456                MOV     #MMP6,R0                ;FSPC
17653  071066  172010                 MM8:   ADDD    (R0),AC0                ;TEST INSTRUCTION
17654  071070  170205                        STFPS   R5                      ;GET FPS
17655  071072  012700  071366                MOV     #MMDAT0,R0              ;GET THE RESULT
17656  071076  174010                        STD     AC0,(R0)
```

```
17657  071100  012701  071436          MOV     #MMP4,R1                ;IS IT CORRECT
17658  071104  012702  000004          MOV     #4,R2
17659  071110  022021          MM9:    CMP     (R0)+,(R1)+
17660  071112  001401                  BEQ     MM10
  (2)  071114  104000                  EMT                             ;
17661  071116  077204          MM10:   SOB     R2,MM9
17662  071120  052704  100004          BIS     #100004,R4
17663  071124  020405                  CMP     R4,R5
17664  071126  001401                  BEQ     MM11
  (2)  071130  104000                  EMT                             ;
17665
17666                          ;THIS IS A TEST OF THE R\T ALGORITHN'S
17667                          ;ABILITY TO SET BOTH N AND Z ON A - 0 RESULT
17668  071132  012704  043200  MM11:   MOV     #043200,R4              ;SET FIV, FIV, AND FD
17669
17670  071136  170104                  LDFPS   R4
17671  071140  012700  071476          MOV     #MMP8,R0                ;SET AC0 OPERAND
17672  071144  172410                  LDD     (R0),AC0
17673  071146  012700  071506          MOV     #MMP9,R0                ;FSPC
17674  071152  172010          MM12:   ADDD    (R0),AC0                ;TEST INSTRUCTION
17675  071154  170205                  STFPS   R5                      ;GET FPS
17676  071156  012700  071366          MOV     #MMDAT0,R0              ;GET THE RESULT
17677  071162  174010                  STD     AC0,(R0)
17678  071164  012701  071466          MOV     #MMP7,R1                ;IS IT CORRECT
17679  071170  012702  000004          MOV     #4,R2
17680  071174  022021          MM13:   CMP     (R0)+,(R1)+
17681  071176  001401                  BEQ     MM16
  (2)  071200  104000                  EMT                             ;
17682  071202  077204          MM16:   SOB     R2,MM13
17683  071204  052704  100014          BIS     #100014,R4              ;FPS CORRECT?
17684  071210  020405                  CMP     R4,R5
17685  071212  001401                  BEQ     MM17
  (2)  071214  104000                  EMT                             ;
17686                          ;TEST THAT CC ARE CLEARED BY R\T
17687  071216  012704  000200  MM17:   MOV     #00200,R4               ;SET FIV, FIV, AND FD
17688  071222  170104                  LDFPS   R4
17689  071224  012700  071476          MOV     #MMP8,R0                ;SET AC0 OPERAND
17690  071230  172410                  LDD     (R0),AC0
17691  071232  012700  071476          MOV     #MMP8,R0                ;FSPC
17692  071236  172010          MM18:   ADDD    (R0),AC0                ;TEST INSTRUCTION
17693  071240  170205                  STFPS   R5                      ;GET FPS
17694  071242  012700  071366          MOV     #MMDAT0,R0              ;GET THE RESULT
17695  071246  174010                  STD     AC0,(R0)
17696  071250  012701  071516          MOV     #MMP10,R1               ;IS IT CORRECT
17697  071254  012702  000004          MOV     #4,R2
17698  071260  022021          MM19:   CMP     (R0)+,(R1)+
17699  071262  001401                  BEQ     MM20
  (2)  071264  104000                  EMT                             ;
17700  071266  077204          MM20:   SOB     R2,MM19
17701  071270  052704  000000          BIS     #00000,R4               ;FPS CORRECT?
17702  071274  020405                  CMP     R4,R5
17703  071276  001401                  BEQ     MM21
  (2)  071300  104000                  EMT                             ;
17704                          ;TEST THAT N IS SET BY R\T
17705  071302  012704  003200  MM21:   MOV     #3200,R4                ;SET FIV, FIV, AND FD
17706  071306  170104                  LDFPS   R4
```

```
17707  071310  012700  071446          MOV     @#HHP5,R0               ;SET ACO OPERAND
17708  071314  172410                  LDD     (R0),AC0
17709  071316  012700  071446          MOV     @#HHP5,R0               ;FSPC
17710  071322  172010          HH22:   ADDD    (R0),AC0                ;TEST INSTRUCTION
17711  071324  170205                  STFPS   R5                      ;GET FPS
17712  071326  012700  071366          MOV     @#HHDAT0,R0             ;GET THE RESULT
17713  071332  174010                  STD     AC0,(R0)
17714  071334  012701  071526          MOV     @#HHP11,R1              ;IS IT CORRECT
17715  071340  012702  000004          MOV     #4,R2
17716  071344  022021          HH23:   CMP     (R0)+,(R1)+
17717  071346  001401                  BEQ     HH24
  (2)  071350  104000                  EMT                             ;
17718  071352  077204          HH24:   SOB     R2,HH23
17719  071354  052704  000010          BIS     #10,R4
17720  071360  020405                  CMP     R4,R5                   ;FPS CORRECT?
17721  071362  001465                  BEQ     HHDONE
  (2)  071364  104000                  EMT                             ;
17722  071366  000000          HHDAT0: 0
17723  071370  000000                  0
17724  071372  000000                  0
17725  071374  000000                  0
17726  071376  000452          HHP0:   452
17727  071400  125252                  125252
17728  071402  125252                  125252
17729  071404  125253                  125253
17730  071406  000252          HHP1:   252
17731  071410  125252                  125252
17732  071412  125252                  125252
17733  071414  125252                  125252
17734  071416  000600          HHP2:   600                             ;HHP0 * HHP1 WITH
17735  071420  000000                  0                               ;PROPER NORMALIZATION
17736  071422  000000                  0
17737  071424  000000                  0
17738  071426  000400          HHP3:   400                             ;HHP0 * HHP1 WITH
17739  071430  000000                  0                               ;BAD NORMALIZATION
17740  071432  000000                  0
17741  071434  000000                  0
17742  071436  000000          HHP4:   0
17743  071440  000000                  0
17744  071442  000000                  0
17745  071444  000000                  0
17746  071446  100200          HHP5:   100200
17747  071450  000000                  0
17748  071452  000000                  0
17749  071454  000000                  0
17750  071456  000300          HHP6:   300
17751  071460  000000                  0
17752  071462  000000                  0
17753  071464  000000                  0
17754  071466  100000          HHP7:   100000                          ;HHF7 = HHP8 * HHP9
17755  071470  000000                  0                               ;      = HHP5 * HHP6
17756  071472  000000                  0
17757  071474  000000                  0
17758  071476  000200          HHP8:   200
17759  071500  000000                  0
17760  071502  000000                  0
```

```
071504  000000                          0
071506  100300                  HHP9:   100300
071510  000000                          0
071512  000000                          0
071514  000000                          0
071516  000400                  HHP10:  400                     ;HHP10 = HHP8 + HHP8
071520  000000                          0
071522  000000                          0
071524  000000                          0
071526  100400                  HHP11:  100400                  ;HHP11 = HHP5 + HHP5
071530  000000                          0
071532  000000                          0
071534  000000                          0
071536                          HHDONE:
071536  004767  033010                  JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
                                                                ;SEE IF THE USER HAS EXPRESSED
                                                                ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                ;THE USER TYPED CONTROL G?).


                                ;*******************************************************************************
                                ;TEST 471       OVER\UNDER TEST
                                ;*******************************************************************************
071542                          TS471:

                                ;TEST OVERFLOW CONDITION WITH TRAP DISABLER FIV=0
071542  012704  000200                  MOV     #200,R4         ;CLEAR FIU, FIV, AND SET FD
071546  170104                          LDFPS   R4
071550  012737  071666  000244          MOV     #GGER0,@#FPVECT
071556  012700  072470                  MOV     #GGP5,R0        ;SET AC0 OPERAND
071562  172410                          LDD     (R0),AC0
071564  012700  072470                  MOV     #GGP5,R0        ;FSRC
071570  172010                  GG2:    ADDD    (R0),AC0        ;TEST INSTRUCTION
071572  170205                          STFPS   R5              ;GET FPS
071574  012700  072420                  MOV     #GGDAT0,R0      ;GET THE RESULT
071600  174010                          STD     AC0,(R0)
071602  012701  072500                  MOV     #GGP6,R1        ;IS IT CORRECT
071606  012702  000004                  MOV     #4,R2
071612  022021                  GG3:    CMP     (R0)+,(R1)+
071614  001401                          BEQ     GG4
071616  104000                          EMT                     ;
071620  077204                  GG4:    SOB     R2,GG3
071622  052704  000006                  BIS     #6,R4           ;FPS CORRECT?
071626  020405                          CMP     R4,R5
071630  001401                          BEQ     GG5
071632  104000                          EMT                     ;
                                        ;TEST OVERFLOW WITH TRAPS ENABLED
                                        ;FIV = 1
071634  012704  001200          GG5:    MOV     #1200,R4        ;CLEAR FIU, SET FIV, AND FD
071640  170104                          LDFPS   R4
071642  012737  071670  000244          MOV     #GG7,@#FPVECT
071650  012700  072470                  MOV     #GGP5,R0        ;SET AC0 OPERAND
071654  172410                          LDD     (R0),AC0
071656  012700  072470                  MOV     #GGP5,R0        ;FSPC
071662  172010                  GG6:    ADDD    (R0),AC0        ;TEST INSTRUCTION
```

```
17820  071664  170000                          CFCC                    ;NO OVERFLOW TRAP OCCURED
17821  071666                          GGERO:
  (2)  071666  104000                          EMT                     ;
17822  071670  012703  071664        GG7:     MOV     #GG6+2,R3
17823  071674  020316                         CMP     R3,(SP)         ;CHECK STACK DATA
17824  071676  001401                         BEQ     1$
  (2)  071700  104000                         EMT                     ;
17825  071702  022626                1$:      CMP     (SP)+,(SP)+
17826  071704  170205                         STFPS   R5
17827  071706  012700  072420                MOV     #GGDAT0,R0      ;GET THE RESULT
17828  071712  174010                         STD     AC0,(R0)
17829  071714  012701  072500                MOV     #GCP6,R1        ;IS IT CORRECT
17830  071720  012702  000004                MOV     #4,R2
17831  071724  022021                GG8:     CMP     (R0)+,(R1)+
17832  071726  001401                         BEQ     GG9
  (2)  071730  104000                         EMT                     ;
17833  071732  077204                GG9:     SOB     R2,GG8
17834  071734  052704  100006                BIS     #100006,R4      ;EXACT ZERO RESULTED IF OVERFLOW
17835  071740  020405                         CMP     R4,R5           ;FPS CORRECT?, CHECK FER, FZ, FV
17836  071742  001401                         BEQ     1$
  (2)  071744  104000                         EMT                     ;
17837  071746  012704  000010        1$:      MOV     #10,R4
17838                                 ;CHECK FEC
17839  071752  170305                         STST    R5
17840  071754  020405                         CMP     R4,R5
17841  071756  001401                         BEQ     GG10
  (2)  071760  104000                         EMT                     ;
17842                                 ;CHECK UNDER FLOW CONDITION WITH
17843                                 ;TRAPS DISABLED (FIU = 0)
17844  071762  012704  000200        GG10:    MOV     #0200,R4        ;SET FIU, FIV, AND FD
17845  071766  170104                         LDFPS   R4
17846  071770  012737  071666  000244        MOV     #GGERO,@#FPVECT
17847  071776  012700  072440                MOV     #GGP2,R0        ;SET AC0 OPERAND
17848  072002  172410                         LDD     (R0),AC0        ;FSRC
17849  072004  012700  072450                MOV     #GGP3,R0
17850  072010  172010                GG11:    ADDD    (R0),AC0        ;TEST INSTRUCTION
17851  072012  170205                         STFPS   R5              ;GET FPS
17852  072014  012700  072420                MOV     #GGDAT0,R0      ;GET THE RESULT
17853  072020  174010                         STD     AC0,(R0)
17854  072022  012701  072500                MOV     #GGP6,R1        ;IS IT CORRECT
17855  072026  012702  000004                MOV     #4,R2
17856  072032  022021                GG12:    CMP     (R0)+,(R1)+
17857  072034  001401                         BEQ     GG13
  (2)  072036  104000                         EMT                     ;
17858  072040  077204                GG13:    SOB     R2,GG12
17859  072042  052704  000004                BIS     #4,R4           ;FPS CORRECT?
17860  072046  020405                         CMP     R4,R5
17861  072050  001401                         BEQ     GG14
  (2)  072052  104000                         EMT                     ;
17862                                 ;CHECK UNDERFLOW CONDITION WITH
17863                                 ;TRAP ENABLED (FIU = 1)
17864  072054  012704  002200        GG14:    MOV     #2200,R4        ;SET FIU, FIV, AND FD
17865  072060  170104                         LDFPS   R4
17866  072062  012737  072106  000244        MOV     #GG16,@#FPVECT
17867  072070  012700  072440                MOV     #GGP2,R0        ;SET AC0 OPERAND
17868  072074  172410                         LDD     (R0),AC0        ;FSPC
```

```
17869  072076  012700  072450          MOV     #GGP3,R0
17870  072102  172010          GG15:   ADDD    (R0),AC0        ;TEST INSTRUCTION
17871  072104  170000                  CFCC
17872  072106  012703  072104  GG16:   MOV     #GG15+2,R3
17873  072112  021603                  CMP     (SP),R3
17874  072114  001401                  BEQ     1$
  (2)  072116  104000                  EMT                     ;
17875  072120  022626          1$:     CMP     (SP)+,(SP)+
17876  072122  170205                  STFPS   R5              ;GET FPS
17877  072124  012700  072420          MOV     #GGDAT0,R0      ;GET THE RESULT
17878  072130  174010                  STD     AC0,(R0)
17879  072132  012701  072510          MOV     #GGP7,R1        ;IS IT CORRECT
17880  072136  012702  000004          MOV     #4,R2
17881  072142  022021          GG17:   CMP     (R0)+,(R1)+
17882  072144  001401                  BEQ     GG18
  (2)  072146  104000                  EMT                     ;
17883  072150  077204          GG18:   SOB     R2,GG17
17884  072152  052704  100000          BIS     #100000,R4
17885  072156  020405                  CMP     R4,R5           ;FPS CORRECT?
17886  072160  001401                  BEQ     1$
  (2)  072162  104000                  EMT                     ;
17887  072164  012704  000012  1$:     MOV     #12,R4
17888                          ;CHECK FEC
17889  072170  170305                  STST    R5
17890  072172  020405                  CMP     R4,R5
17891  072174  001401                  BEQ     GG19
  (2)  072176  104000                  EMT                     ;
17892                          ;CHECK UNDERFLOW CONDITION WITH TRAPS
17893                          ;DISABLED (FIU = 0)
17894  072200  012704  000200  GG19:   MOV     #0200,R4        ;SET FIU, FIV, AND FD
17895  072204  170104                  LDFPS   R4
17896  072206  012737  072324  000244  MOV     #GGER14,@#FPVECT
17897  072214  012700  072440          MOV     #GGP2,R0        ;SET AC0 OPERAND
17898  072220  172410                  LDD     (R0),AC0
17899  072222  012700  072520          MOV     #GGP8,R0        ;FSPC
17900  072226  172010          GG20:   ADDD    (R0),AC0        ;TEST INSTRUCTION
17901  072230  170205                  STFPS   R5              ;GET FPS
17902  072232  012700  072420          MOV     #GGDAT0,R0      ;GET THE RESULT
17903  072236  174010                  STD     AC0,(R0)
17904  072240  012701  072500          MOV     #GGP6,R1        ;IS IT CORRECT
17905  072244  012702  000004          MOV     #4,R2
17906  072250  022021          GG21:   CMP     (R0)+,(R1)+
17907  072252  001401                  BEQ     GG22
  (2)  072254  104000                  EMT                     ;
17908  072256  077204          GG22:   SOB     R2,GG21
17909  072260  052704  000004          BIS     #4,R4           ;FPS CORRECT?
17910  072264  020405                  CMP     R4,R5
17911  072266  001401                  BEQ     GG23
  (2)  072270  104000                  EMT                     ;
17912                                  ;CHECK UNDERFLOW CONDITION WITH TRAP
17913                                  ;ENABLED (FIU = 1)
17914  072272  012704  002200  GG23:   MOV     #2200,R4        ;SET FIU, FIV, AND FD
17915  072276  170104                  LDFPS   R4
17916  072300  012737  072326  000244  MOV     #GG25,@#FPVECT
17917  072306  012700  072440          MOV     #GGP2,R0        ;SET AC0 OPERAND
17918  072312  172410                  LDD     (R0),AC0
```

```
17919  072314  012700  072520          MOV     #GGP8,R0        ;FSRC
17920  072320  172010          GG24:   ADDD    (R0),AC0        ;TEST INSTRUCTION
17921  072322  170000                  CFCC
17922  072324                  GGER14:
  (2)  072324  104000                  EMT                     ;
17923  072326  012703  072322  GG25:   MOV     #GG24+2,R3
17924  072332  020316                  CMP     R3,(SP)
17925  072334  001401                  BEQ     1$
  (2)  072336  104000                  EMT                     ;
17926  072340  022626          1$:     CMP     (SP)+,(SP)+
17927  072342  170205                  STFPS   R5              ;GET FPS
17928  072344  012700  072420          MOV     #GGDAT0,R0      ;GET THE RESULT
17929  072350  174010                  STD     AC0,(R0)
17930  072352  012701  072530          MOV     #GGP9,R1        ;IS IT CORRECT
17931  072356  012702  000004          MOV     #4,R2
17932  072362  022021          GG26:   CMP     (R0)+,(R1)+
17933  072364  001401                  BEQ     GG27
  (2)  072366  104000                  EMT                     ;
17934  072370  077204          GG27:   SOB     R2,GG26
17935  072372  052704  100004          BIS     #100004,R4
17936  072376  020405                  CMP     R4,R5           ;FPS CORRECT?
17937  072400  001401                  BEQ     1$
  (2)  072402  104000                  EMT                     ;
17938  072404  012704  000012  1$:     MOV     #12,R4
17939                                  ;CHECK FEC
17940  072410  170305                  STST    R5
17941  072412  020405                  CMP     R4,R5
17942  072414  001451                  BEQ     GGDONE
  (2)  072416  104000                  EMT                     ;
17943  072420  000000          GGDATO: 0
17944  072422  000000                  0
17945  072424  000000                  0
17946  072426  000000                  0
17947
17948  072430  000300          GGP1:   300
17949  072432  000000                  0
17950  072434  000000                  0
17951  072436  000000                  0
17952  072440  100200          GGP2:   100200
17953  072442  000000                  0
17954  072444  000000                  0
17955  072446  000000                  0
17956  072450  000200          GGP3:   200
17957  072452  000000                  0
17958  072454  000000                  0
17959  072456  000001                  1
17960  072460  010200          GGP4:   10200
17961  072462  000000                  C
17962  072464  000000                  0
17963  072466  000000                  0
17964  072470  077600          GGP5:   77600                   ;OVER FLOW = GGP5 + GGP5
17965  072472  000000                  0
17966  072474  000000                  0
17967  072476  000000                  0
17968  072500  000000          GGP6:   0                       ;OVERFLOW RESULT
17969  072502  000000                  0                       ;UNDERFLOW RESULT
```

```
17970  072504  000000                    0                       ;GGP6 = GGP4 + GGP5
17971  072506  000000                    0                       ;      = GGP3 + GGP2 (FIU = 0)
17972                                                            ;      = GGP3 + GGP1
17973  072510  062400            GGP7:   62400                   ;GGP7 = GGP3 + GGP2 (FIU = 1)
17974  072512  000000                    0
17975  072514  000000                    0
17976  072516  000000                    0
17977  072520  000340            GGP8:   340
17978  072522  000000                    0
17979  072524  000000                    0
17980  072526  000000                    0
17981  072530  000100            GGP9:   100
17982  072532  000000                    0
17983  072534  000000                    0
17984  072536  000000                    0
17985  072540                    GGDONE:
  (1)  072540  004767  032006            JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                            ;SEE IF THE USER HAS EXPRESSED
  (1)                                                            ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                            ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                            ;THE USER TYPED CONTROL G?).
17986
17992
17993                             ;*************************************************************************************
  (2)                             ;TEST 472       LDCFD AND LDCDF TEST
  (3)                             ;*************************************************************************************
  (2)  072544                     TS472:
17994                             ;TEST FOR CORRECT AUTO INCREMENT CONSTANT.
17995  072544  012704  000200            MOV     #200,R4         ;SET LONG INTEGER MODE
17996  072550  170104                    LDFPS   R4
17997  072552  012700  073302            MOV     #HXP1,R0
17998  072556  172410                    LDD     (R0),AC0
17999  072560  012700  073312            MOV     #HXP2,R0
18000  072564  177420            HX2:    LDCFD   (R0)+,AC0
18001  072566  020027  073316            CMP     R0,#HXP2+4      ;IS R0 CORRECT
18002  072572  001401                    BEQ     HX3
  (2)  072574  104000                    EMT                     ;
18003  072576                    HX3:
18004  072576  170205                    STFPS   R5              ;GET FPS
18005  072600  012700  073272            MOV     #HXDAT0,R0
18006  072604  174010                    STD     AC0,(R0)        ;GET AC0
18007  072606  012701  073362            MOV     #HXP7,R1        ;SEE IF RESULT IS
18008  072612  012702  000004            MOV     #4,R2           ;CORRECT
18009  072616  022120            HX4:    CMP     (R1)+,(R0)+
18010  072620  001401                    BEQ     HX7
  (2)  072622  104000                    EMT                     ;
18011  072624  077204            HX7:    SOB     R2,HX4
18012  072626  012704  000200            MOV     #200,R4         ;FPS CORRECT?
18013  072632  020405                    CMP     R4,R5
18014  072634  001401                    BEQ     HX8
  (2)  072636  104000                    EMT                     ;
18015                             ;NOW    TEST    LDCDF
18016  072640  012704  000200    HX8:    MOV #200,R4
18017  072644  170104                    LDFPS R4
18018
18019  072646  012700  073302            MOV     #HXP1,R0
```

```
18020  072652  172410                        LDD     (R0),AC0
18021
18022  072654  012700  073312                MOV     #HXP2,R0
18023  072660  170001                        SETF
18024  072662  177420                HX9:    LDCDF   (R0)+,AC0       ;TEST INSTRUCTION
18025  072664  020027  073322                CMP     R0,#HXP2+10     ;WAS A GOOD
18026  072670  001401                        BEQ     HX10
  (2)  072672  104000                        EMT                     ;
18027
18028  072674                         HX10:
18029  072674  170205                        STFPS   R5
18030  072676  012700  073272                MOV     #HXDAT0,R0
18031  072702  170011                        SETD
18032  072704  174010                        STD     AC0,(R0)        ;GET RESULT
18033  072706  012701  073372                MOV     #HXP8,R1
18034  072712  012702  000004                MOV     #4,R2
18035  072716  022120                HX11:   CMP     (R1)+,(R0)+     ;IS IT CORRECT?
18036  072720  001401                        BEQ     HX14
  (2)  072722  104000                        EMT                     ;
18037  072724  077204                HX14:   SOB     R2,HX11
18038
18039  072726  012704  000000                MOV     #0,R4           ;FPS CORRECT?
18040  072732  020405                        CMP     R4,R5
18041  072734  001401                        BEQ     HX15
  (2)  072736  104000                        EMT                     ;
18042                                 ;TEST GR7 IMMEDIATE MODE CONSTANT
18043  072740  012704  000200        HX15:   MOV     #200,R4
18044  072744  170104                        LDFPS   R4              ;SET FD
18045  072746  012737  072776  000004        MOV     #HXER9,@#ERRVECT
18046  072754  005001                        CLR     R1
18047  072756  177427  043243        HX16:   LDCFD   #5201,AC0
18048  072762  005201                HX165:  INC     R1
18049  072764  005201                        INC     R1
18050  072766  005201                        INC     R1
18051  072770  020127  000003                CMP     R1,#3           ;SEE IF PC WAS
18052  072774  001401                        BEQ     HX17
  (1)  072776                         HXER9:
  (2)  072776  104000                        EMT                     ;
18053  073000  012704  000200        HX17:   MOV     #200,R4
18054  073004  170104                        LDFPS   R4
18055  073006  012700  073352                MOV     #HXP6,R0
18056  073012  172410                        LDD     (R0),AC0
18057  073014  012700  073312                MOV     #HXP2,R0
18058  073020  177410                HX18:   LDCFD   (R0),AC0
18059
18060  073022  012700  073272                MOV     #HXDAT0,R0
18061  073026  174010                        STD     AC0,(R0)        ;GET RESULT.
18062  073030  012701  073362                MOV     #HXP7,R1
18063  073034  012702  000004                MOV     #4,R2
18064  073040  022021                HX19:   CMP     (R0)+,(R1)+     ;IS RESULT CORRECT?
18065  073042  001401                        BEQ     HX20
  (2)  073044  104000                        EMT                     ;
18066  073046  077204                HX20:   SOB     R2,HX19
18067
18068                                 ;TEST LDCFD WITH NEGATIVE OPERAND
18069  073050  012704  000200                MOV     #200,R4
```

```
18070  073054  170104                        LDFPS   R4
18071  073056  012700  073352                MOV     #HXP6,R0
18072  073062  172410                        LDD     (R0),AC0
18073  073064  012700  073332                MOV     #HXP4,R0
18074  073070  177410              HX22:     LDCFD   (R0),AC0
18075
18076  073072  012700  073272                MOV     #HXDAT0,R0
18077  073076  174010                        STD     AC0,(R0)        ;GET RESULT
18078
18079  073100  012701  073342                MOV     #HXP5,R1
18080  073104  012702  000004                MOV     #4,R2
18081  073110  022120              HX23:     CMP     (R1)+,(R0)+
18082  073112  001401                        BEQ     HX26
  (2)  073114  104000                        EMT                     ;
18083  073116  077204              HX26:     SOB     R2,HX23
18084
18085                               ;TEST     LDCFD   0
18086
18087  073120  012704  000200                MOV     #200,R4
18088  073124  170104                        LDFPS   R4
18089
18090  073126  012700  073302                MOV     #HXP1,R0
18091  073132  172410                        LDD     (R0),AC0
18092  073134  172010                        ADDD    (R0),AC0
18093
18094  073136  012700  073302                MOV     #HXP1,R0
18095  073142  177410              HX28:     LDCFD   (R0),AC0
18096
18097  073144  170205                        STFPS   R5
18098
18099  073146  012700  073272                MOV     #HXDAT0,R0
18100  073152  174010                        STD     AC0,(R0)        ;GET RESULT
18101
18102  073154  012701  073302                MOV     #HXP1,R1
18103  073160  012702  000004                MOV     #4,R2
18104  073164  022120              HX29:     CMP     (R1)+,(R0)+     ;IS IT 0?
18105  073166  001401                        BEQ     HX30
  (2)  073170  104000                        EMT                     ;
18106  073172  077204              HX30:     SOB     R2,HX29
18107
18108  073174  012704  000204                MOV     #204,R4         ;FPS CORRECT
18109  073200  020405                        CMP     R4,R5
18110  073202  001401                        BEQ     HX31
  (2)  073204  104000                        EMT                     ;
18111                               ;TEST     LDCFD   0
18112  073206  012704  000200      HX31:     MOV     #200,R4
18113  073212  170104                        LDFPS   R4
18114  073214  012700  073352                MOV     #HXP6,R0
18115  073220  172410                        LDD     (R0),AC0
18116  073222  012700  073302                MOV     #HXP1,R0
18117  073226  177410              HX32:     LDCFD   (R0),AC0
18118  073230  170205                        STFPS   R5
18119  073232  012700  073272                MOV     #HXDAT0,R0
18120  073236  174010                        STD     AC0,(R0)        ;GET RESULT
18121  073240  012701  073302                MOV     #HXP1,R1
18122  073244  012702  000004                MOV     #4,R2
```

```
18123  073250  022120                HX33:   CMP     (R1)+,(R0)+     ;IS IT ZERO?
18124  073252  001401                        BEQ     HX34
  (2)  073254  104000                        EMT                     ;
18125  073256  077204                HX34:   SOB     R2,HX33
18126
18127  073260  012704  000204                MOV     #204,R4         ;FPS CORRECT?
18128  073264  020405                        CMP     R4,R5
18129  073266  001445                        BEQ     HXDONE
  (2)  073270  104000                        EMT                     ;
18130
18131  073272  000000                HXDATO: 0
18132  073274  000000                        0
18133  073276  000000                        0
18134  073300  000000                        0
18135
18136  073302  000000                HXP1:   0
18137  073304  000000                        0
18138  073306  000000                        0
18139  073310  000000                        0
18140
18141  073312  000577                HXP2:   577
18142  073314  177776                        177776
18143  073316  177777                        177777
18144  073320  177776                        177776
18145  073322  005201                HXP3:   5201
18146  073324  000000                        0
18147  073326  000000                        0
18148  073330  000000                        0
18149  073332  100577                HXP4:   100577
18150  073334  177776                        177776
18151  073336  177777                        177777
18152  073340  177776                        177776
18153  073342  100577                HXP5:   100577
18154  073344  177776                        177776
18155  073346  000000                        0
18156  073350  000000                        0
18157  073352  000252                HXP6:   252
18158  073354  125252                        125252
18159  073356  125252                        125252
18160  073360  125252                        125252
18161
18162  073362  000577                HXP7:   577
18163  073364  177776                        177776
18164  073366  000000                        0
18165  073370  000000                        0
18166  073372  000577                HXP8:   577
18167  073374  177777                        177777
18168  073376  000000                        0
18169  073400  000000                        0
18170
18171  073402                        HXDONE:
  (1)  073402  004767  031144                JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                ;THE USER TYPED CONTROL G?).
```

```
                                        ;*******************************************************************************
                                        ;TEST 473        CMPD TEST
                                        ;*******************************************************************************
073406                                  TS473:

                                        ;TEST THE CMPD INSTRUCTION WITH (FSRC=AC=0)
073406  004737  074102                  AAA1:   JSR     PC,@#CMPSUB
073412  000000  000000  000000          1$:     .WORD   0,0,0,0                 ;AC0
073420  000000
073422  000000  000000  000000          2$:     .WORD   0,0,0,0                 ;FSRC
073430  000000
073432  000200                          3$:     200                             ;FPS BEFORE EXECUTION
073434  000204                                  204                             ;FPS AFTER EXECUTION


                                        ;TEST CMPD WITH (AC=0) AND FSRC POSITIVE.
073436  004737  074102                  AAA2:   JSR     PC,@#CMPSUB
073442  000000  000000  000000          1$:     .WORD   0,0,0,0                 ;AC
073450  000000
073452  025252                          2$:     25252                           ;FSRC
073454  052525                                  52525
073456  125252                                  125252
073460  052525                                  52525
073462  000200                          3$:     200                             ;FPS BEFORE EXECUTION
073464  000200                                  200                             ;FPS AFTER EXECUTION

                                        ;TEST CMPD WITH (AC=0) AND FSRC NEGATIVE
073466  004737  074102                  AAA3:   JSR     PC,@#CMPSUB
073472  000000  000000  000000          1$:     .WORD   0,0,0,0                 ;AC
073500  000000
073502  125252                          2$:     125252                          ;FSRC
073504  125252                                  125252
073506  052525                                  52525
073510  125252                                  125252
073512  000200                          3$:     200                             ;FPS BEFORE EXECUTION
073514  000210                                  210                             ;FPS AFTER EXECUTION

                                        ;TEST CMPD WITH (FSRC=0) AND AC POSITIVE
073516  004737  074102                  AAA4:   JSR     PC,@#CMPSUB
073522  025252                          1$:     25252                           ;AC
073524  052525                                  52525
073526  125252                                  125252
073530  052525                                  52525
073532  000000  000000  000000          2$:     .WORD   0,0,0,0                 ;FSRC
073540  000000
073542  000200                          3$:     200                             ;FPS BEFORE EXECUTION
073544  000210                                  210                             ;FPS AFTER EXECUTION


                                        ;TEST CMPD WITH (FSRC=0) AND AC NEGATIVE
073546  004737  074102                  AAA5:   JSR     PC,@#CMPSUB
073552  125252                          1$:     125252                          ;AC
```

```
18227  073554  125252                          125252
18228  073556  052525                          52525
18229  073560  125252                          125252
18230  073562  000000  000000  000000  2$:     .WORD   0,0,0,0                 ;FSRC
       073570  000000
18231  073572  000200                  3$:     200                             ;FPS BEFORE EXECUTION
18232  073574  000200                          200                             ;FPS AFTER EXECUTION
18233
18234                                  ;TEST CMPD WITH AC POSITIVE AND FSRC NEGATIVE
18235  073576  004737  074102          AAA6:   JSR     PC,@#CMPSUB
18236  073602  052525                  1$:     52525                           ;AC
18237  073604  125252                          125252
18238  073606  052525                          52525
18239  073610  125252                          125252
18240  073612  125252                  2$:     125252                          ;;FSRC
18241  073614  052525                          52525
18242  073616  125252                          125252
18243  073620  052525                          52525
18244  073622  000200                  3$:     200                             ;FPS BEFORE EXECUTION
18245  073624  000210                          210                             ;FPS AFTER EXECUTION
18246
18247
18248                                  ;TEST CMPD WITH AC NEGATIVE AND FSRC POSITIVE
18249  073626  004737  074102          AAA7:   JSR     PC,@#CMPSUB
18250  073632  125252                  1$:     125252                          ;AC
18251  073634  052525                          52525
18252  073636  125252                          125252
18253  073640  052525                          52525
18254  073642  052525                  2$:     52525                           ;FSRC
18255  073644  125252                          125252
18256  073646  052525                          52525
18257  073650  125252                          125252
18258  073652  000200                  3$:     200                             ;FPS BEFORE EXECUTION
18259  073654  000200                          200                             ;FPS AFTER EXECUTION
18260
18261                                  ;TEST CMPD WITH AC POSITIVE AND FSRC POSITIVE
18262                                  ;AND EAC LESS THAN EFSRC.
18263  073656  004737  074102          AAA8:   JSR     PC,@#CMPSUB
18264  073662  012345                  1$:     12345                           ;AC
18265  073664  067654                          67654
18266  073666  032101                          32101
18267  073670  023456                          23456
18268  073672  023456                  2$:     23456                           ;FSRC
18269  073674  076543                          76543
18270  073676  021012                          21012
18271  073700  034567                          34567
18272  073702  000200                  3$:     200                             ;FPS BEFORE EXECUTION
18273  073704  000200                          200                             ;FPS AFTER EXECUTION
18274
18275
18276                                  ;TEST CMPD WITH AC POSITIVE, FSRC POSITIVE AND EAC GREATER THAN EFSRC
18277  073706  004737  074102          AAA9:   JSR     PC,@#CMPSUB
18278  073712  045676                  1$:     45676                           ;AC
18279  073714  054321                          54321
18280  073716  012345                          12345
18281  073720  067654                          67654
```

```
18282  073722  034567                  2$:     34567                           ;FSRC
18283  073724  065432                          65432
18284  073726  101234                          101234
18285  073730  056765                          56765
18286  073732  000200                  3$:     200                             ;FPS BEFORE EXECUTION
18287  073734  000210                          210                             ;FPS AFTER EXECUTION
18288
18289                                  ;TEST CMPD WITH AC POSITIVE, FSRC POSITIVE AND AC EQUAL TO FSRC
18290  073736  004737  074102          AAA10:  JSR     PC,@#CMPSUB
18291  073742  012345                  1$:     12345                           ;AC
18292  073744  067012                          67012
18293  073746  034567                          34567
18294  073750  012345                          012345
18295  073752  012345                  2$:     12345                           ;FSRC
18296  073754  067012                          67012
18297  073756  034567                          34567
18298  073760  012345                          012345
18299  073762  000200                  3$:     200                             ;FPS BEFORE EXECUTION
18300  073764  000204                          204                             ;FPS AFTER EXECUTION
18301
18302                                  ;TEST CMPD WITH AC POSITIVE, FSRC POSITIVE, EAC EQUAL TO EFSRC,
18303                                  ;AND FSRC GREATER THAN AC.
18304  073766  004737  074102          AAA11:  JSR     PC,@#CMPSUB
18305  073772  012345                  1$:     12345                           ;AC
18306  073774  067012                          67012
18307  073776  034567                          34567
18308  074000  012345                          012345
18309  074002  012345                  2$:     12345                           ;FSRC
18310  074004  070123                          70123
18311  074006  045670                          45670
18312  074010  123456                          123456
18313  074012  000200                  3$:     200                             ;FPS BEFORE EXECUTION
18314  074014  000200                          200                             ;FPS AFTER EXECUTION
18315
18316                                  ;TEST CMPD WITH AC POSITIVE, FSRC POSITIVE, EAC EQUAL TO EFSRC,
18317                                  ;AND AC GREATER THAN FSRC.
18318  074016  004737  074102          AAA12:  JSR     PC,@#CMPSUB
18319  074022  054321                  1$:     54321                           ;AC
18320  074024  076543                          76543
18321  074026  021076                          21076
18322  074030  054321                          54321
18323  074032  054321                  2$:     54321                           ;FSRC
18324  074034  065432                          65432
18325  074036  107654                          107654
18326  074040  032107                          32107
18327  074042  000200                  3$:     200                             ;FPS BEFORE EXECUTION
18328  074044  000210                          210                             ;FPS AFTER EXECUTION
18329
18330                                  ;TEST CMPD WITH AC NEGATIVE, FSRC NEGATIVE, EAC EQUAL TO EFSRC,
18331                                  ;AND AC GREATER THAN FSRC
18332  074046  004737  074102          AAA13:  JSR     PC,@#CMPSUB
18333  074052  112345                  1$:     112345                          ;AC
18334  074054  043210                          43210
18335  074056  076543                          76543
18336  074060  021076                          21076
18337  074062  112345                  2$:     112345                          ;FSRC
```

```
18338  074064  054321                    54321
18339  074066  007654                    07654
18340  074070  032107                    32107
18341  074072  000200              3$:   200                     ;FPS BEFORE EXECUTION
18342  074074  000210                    210                     ;FPS AFTER EXECUTION
18343
18344
18345  074076  000137  074206            JMP     @@AAADONE       ;FINISHED CMPD TEST.
18346
18347
18348                                    ;THIS SUBROUTINE, CMPSUB, IS CALLED TO SET UP, EXECUTE
18349                                    ;AND CHECK THE RESULTS OF A CMPD INSTRUCTION.
18350                                    ;IT IS CALLED THUS:
18351                                    ;
18352                                    ;       JSR     PC,@@CMPSUB
18353                                    ;ACARG:  .WORD   X,X,X,X         ;AC OPERAND
18354                                    ;FSRCARG:.WORD   X,X,X,X         ;FSRC OPERAND
18355                                    ;FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
18356                                    ;FPSA:   .WORD   X               ;FPS AFTER EXECUTION
18357                                    ;FPSE:   .WORD   X               ;ERROR FPS
18358                                    ;ERR:    ERROR   X               ;FPS ERROR
18359                                    ;CONT:                           ;RETURN ADDRESS
18360                                    ;
18361                                    ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC OPERAND). THEN
18362                                    ;FPSB IS LOADED INTO THE FPS. THE INSTRUCTION, CMPD, IS EXECUTED.
18363                                    ;AFTER THE EXECUTION THE FPS IS CHECKED AGAINST FPSA. IF IT IS A MATCH
18364                                    ;THEN THERE WAS NO ERROR AND CONTROL IS RETURNED TO CONT. IF
18365                                    ;THE FPS IS INCORRECT IT IS COMPARED WITH FPSE IN AN ATTEMPT TO ANALYSE
18366                                    ;THE FAILURE. IF THE FPS IS THE SAME AS FPSE THEN CONTROL IS
18367                                    ;RETURNED TO THE ERROR CALL AT LOCATION ERR. IF THE FPS WAS
18368                                    ;NOT CORRECT BUT DIDN'T MATCH FPSE A GENERAL ERROR IS REPORTED
18369                                    ;AND CONTROL IS PASSED TO CONT.
18370
18371  074102  012601            CMPSUB: MOV     (SP)+,R1        ;PICK UP A POINTER TO THE
18372                                                             ;ARGUMENTS.
18373  074104  016100  000020            MOV     20(R1),R0       ;GET THE FPS BEFORE EXECUTION.
18374  074110  170100                    LDFPS   R0              ;LOAD IT INTO THE FPS.
18375
18376  074112  010100                    MOV     R1,R0           ;GET ADDRESS OF AC OPERAND.
18377  074114  172410                    LDD     (R0),AC0        ;LOAD AC0 OPERAND
18378
18379  074116  010100                    MOV     R1,R0           ;COMPUTE FSRC OPERAND
18380  074120  062700  000010            ADD     #10,R0          ;ADDRESS
18381
18382  074124  000240                    NOP                     ;FOR SCOPING.
18383  074126  173410              1$:   CMPD    (R0),AC0        ;EXECUTE THE TEST INSTRUCTION.
18384
18385  074130  170205                    STFPS   R5              ;SAVE FPS AFTER INSTRUCTION.
18386
18387  074132  016104  000022            MOV     22(R1),R4       ;GET EXPECTED FPS.
18388  074136  020405                    CMP     R4,R5           ;WAS FPS CORRECT?
18389  074140  020405                    CMP     R4,R5           ;WAS FPS CORRECT?
18390  074142  001401                    BEQ     3$
  (2)  074144  104000                    EMT                     ;
18391  074146  012700  074176      3$:   MOV     #CMPTMP,R0      ;IF FPS WAS CORRECT MAKE SURE
18392  074152  174010                    STD     AC0,(R0)        ;AC0 WAS NOT AFFECTED BY CMPD.
```

```
18393  074154  010102                        MOV     R1,R2
18394  074156  012703  000004                MOV     #4,R3
18395  074162  022220                4$:     CMP     (R2)+,(R0)+
18396  074164  001401                        BEQ     5$
  (2)  074166  104000                        EMT                     ;
18397  074170  077304                5$:     SOB     R3,4$
18398
18399  074172  000161  000024                JMP     24(R1)                  ;RETURN
18400
18401  074176  000000  000000  000000  CMPTMP: .WORD  0,0,0,0
       074204  000000
18402
18403
18404
18405  074206                        AAADONE:
  (1)  074206  004767  030340                JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                               ;SEE IF THE USER HAS EXPRESSED
  (1)                                                               ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                               ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                               ;THE USER TYPED CONTROL G?).
18406
18407
18408
18416
18417                           ;*************************************************************************************
  (2)                           ;TEST 474       DIVD WITH (FSRC=0) AND (BUT FD) TEST
  (3)                           ;*************************************************************************************
  (2)  074212                   TS474:
18418
18419  074212  012704  040200   BBB0:   MOV     #40200,R4       ;SET UP FPS
18420                                                             ;WITH INTERRUPTS
18421                                                             ;DISABLED.
18422  074216  170104                   LDFPS   R4
18423  074220  012737  074256  000244   MOV     #BBBER1,@#FPVECT;SET UP FOR ANY FP INTERRUPTS.
18424  074226  012700  074442           MOV     #BBBP1,R0       ;SET UP AC0 = 0
18425  074232  172410                   LDD     (R0),AC0
18426  074234  012701  074442           MOV     #BBBP1,R1       ;FSRC = 0
18427
18428  074240  174411           BBB1:   DIVD    (R1),AC0        ;TEST INSTRUCTION
18429
18430  074242  170205                   STFPS   R5              ;GET FPS
18431  074244  170303                   STST    R3              ;GET FEC
18432
18433  074246  012704  140204           MOV     #140204,R4      ;EXPECTED FPS.
18434  074252  020405                   CMP     R4,R5           ;IS FPS CORRECT.
18435  074254  001401                   BEQ     BBB7
  (1)  074256                   BBBER1:
  (2)  074256  104000                   EMT                     ;
18436  074260  012702  000004   BBB7:   MOV     #4,R2           ;EXPECTED FEC.
18437  074264  020203                   CMP     R2,R3           ;IS FEC CORRECT?
18438  074266  001401                   BEQ     BBB2
  (2)  074270  104000                   EMT                     ;
18439
18440                           ;TEST DIVD WITH (FSRC=0) AND TRAPS DISABLED.
18441  074272  012704  040200   BBB2:   MOV     #40200,R4       ;LOAD FPS WITH TRAPS DISABLED.
18442  074276  170104                   LDFPS   R4
```

```
18443
18444  074300  012700  074452          MOV     #BBBP2,R0       ;SET UP AC0 OPERAND (NON ZERO).
18445  074304  172410                  LDD     (R0),AC0
18446  074306  012700  074442          MOV     #BBBP1,R0       ;FSRC=0
18447  074312  174410          BBB3:   DIVD    (R0),AC0
18448
18449  074314  170205                  STFPS   R5              ;GET FPS.
18450  074316  170303                  STST    R3              ;GET FEC.
18451
18452  074320  012704  140200          MOV     #140200,R4              ;EXPECTED FPS.
18453  074324  020405                  CMP     R4,R5           ;IS FPS CORRECT?
18454  074326  001401                  BEQ     1$
  (2)  074330  104000                  EMT                     ;
18455  074332  012702  000004  1$:     MOV     #4,R2           ;EXPECTED FEC.
18456  074336  020203                  CMP     R2,R3           ;WAS FEC CORRECT?
18457  074340  001401                  BEQ     BBB4
  (2)  074342  104000                  EMT                     ;
18458
18459                          ;TEST DIVD WITH FSRC=0) AND TRAPS ENABLED.
18460  074344  012704  000200  BBB4:   MOV     #200,R4         ;SET UP FPS. TRAP ENABLED.
18461  074350  170104                  LDFPS   R4
18462
18463  074352  012700  074452          MOV     #BBBP2,R0       ;SET UP AC0 OPERAND (NON ZERO).
18464  074356  172410                  LDD     (R0),AC0
18465
18466  074360  012737  074400  000244  MOV     #BBB6,@#FPVECT  ;SET UP FOR THE EXPECTED INTERRUPT.
18467  074366  012700  074442          MOV     #BBBP1,R0       ;FSRC=0
18468
18469  074372  174410          BBB5:   DIVD    (R0),AC0        ;TEST INSTRUCTION (SHOULD RESULT IN TRAP).
18470  074374  170000                  CFCC
18471  074376  104000                  EMT                     ;
18472  074400  022716  074374  BBB6:   CMP     #BBB5+2,(SP)    ;TRAP TO HERE WHEN THE DIVISION BY 0
18473                                  ;OCCURS. FIRST SEE IF THE ADDRESS OF
18474                                  ;THE TRAP IS 2+THE ADDRESS OF THE TEST
18475                                  ;DIVD INSTRUCTION.
18476  074404  001401                  BEQ     1$
  (2)  074406  104000                  EMT                     ;
18477  074410  170205          1$:     STFPS   R5              ;GET FPS.
18478  074412  170303                  STST    R3              ;GET FEC.
18479  074414  022626                  CMP     (SP)+,(SP)+     ;RESET THE STACK.
18480
18481  074416  012704  100200          MOV     #100200,R4      ;EXPECTED FPS.
18482  074422  020405                  CMP     R4,R5           ;IS FPS CORRECT?
18483  074424  001401                  BEQ     2$
  (2)  074426  104000                  EMT                     ;
18484  074430  012702  000004  2$:     MOV     #4,R2           ;EXPECTED FEC.
18485  074434  020203                  CMP     R2,R3           ;IS FEC CORRECT?
18486  074436  001411                  BEQ     BBBDONE
  (2)  074440  104000                  EMT                     ;
18487
18488  074442  000000  000000  000000  BBBP1:  .WORD   0,0,0,0
       074450  000000
18489  074452  012345  054321  023456  BBBP2:  .WORD   12345,54321,23456,76543
       074460  076543
18490
18491
```

```
18492
18493  074462                          BBBDONE:
  (1)  074462  004767  030064                JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                 ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                 ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                 ;THE USER TYPED CONTROL G?).
18494
18495
18503
18504                                 ;*****************************************************************************
  (2)                                 ;TEST 475       DIVF TEST
  (3)                                 ;*****************************************************************************
  (2)  074466                         TS475:
18505
18506                                 ;CHECK DIVF WITH (AC=0).
18507  074466  004767  000404         CCC1:   JSR     PC,DIVFSUB
18508  074472  000000  000000         1$:     .WORD   0,0             ;AC
18509  074476  012345  067012         2$:     .WORD   12345,67012     ;FSRC
18510  074502  000000  000000         3$:     .WORD   0,0             ;RES
18511  074506  000000                 4$:     0                       ;FPS BEFORE EXECUTION
18512  074510  000004                         4                       ;FPS AFTER EXECUTION
18513
18514                                 ;TEST DIVF WITH AC POSITIVE, FSRC POSITIVE AND IN ROUND MODE.
18515  074512  004737  075076         CCC2:   JSR     PC,@#DIVFSUB
18516  074516  065652  125252         1$:     .WORD   65652,125252    ;AC
18517  074522  065600  000000         2$:     .WORD   65600,0         ;FSRC
18518  074526  040252  125252         3$:     .WORD   40252,125252    ;RES
18519  074532  003000                 4$:     3000                    ;FPS BEFORE EXECUTION.
18520  074534  003000                         3000                    ;FPS AFTER EXECUTION.
18521
18522                                 ;TEST DIVF WITH AC POSITIVE, FSRC POSITIVE.
18523  074536  004767  000334         CCC3:   JSR     PC,DIVFSUB
18524  074542  076400  000000         1$:     .WORD   76400,0         ;AC
18525  074546  076400  000000         2$:     .WORD   76400,0         ;FSRC
18526  074552  040200  000000         3$:     .WORD   40200,0         ;RES
18527  074556  001000                 4$:     1000                    ;FPS BEFORE EXECUTION.
18528  074560  001000                         1000                    ;FPS AFTER EXECUTION.
18529                                 ;TEST DIVF WITH BOTH OPERANDS POSITIVE.
18530  074562  004737  075076         CCC4:   JSR     PC,@#DIVFSUB
18531  074566  056777  177777         1$:     .WORD   56777,177777    ;AC
18532  074572  054200  000000         2$:     .WORD   54200,0         ;FSRC
18533  074576  042777  177777         3$:     .WORD   42777,177777    ;RES
18534  074602  000000                 4$:     0                       ;FPS BEFORE EXECUTION.
18535  074604  000000                         0                       ;FPS AFTER EXECUTION.
18536
18537                                 ;TEST THE DIVF INSTRUCTION:
18538  074606  004737  075076         CCC5:   JSR     PC,@#DIVFSUB
18539  074612  012377  177777         1$:     .WORD   12377,177777    ;AC
18540  074616  012300  000000         2$:     .WORD   12300,0         ;FSRC
18541  074622  040252  125252         3$:     .WORD   40252,125252    ;RES
18542  074626  000000                 4$:     0                       ;FPS BEFORE EXECUTION.
18543  074630  000000                         0                       ;FPS AFTER EXECUTION.
18544
18545                                 ;TEST DIVIDE ALGORITHM. TEST ROUND CONSTANT.
18546  074632  004737  075076         CCC6:   JSR     PC,@#DIVFSUB
```

```
 18547  074636  064600  000001          1$:     .WORD   64600,1         ;AC
```

```
18549  074642  066600  000000          2$:     .WORD   66600.0          ;FSRC
18550  074646  036200  000001          3$:     .WORD   36200.1          ;RES
18551  074652  000000                  4$:     0                        ;FPS BEFORE EXECUTION.
18552  074654  000000                          0                        ;FPS AFTER EXECUTION.
18553
18554                                  ;TEST DIVF.
18555  074656  004737  075076          CCC7:   JSR     PC,@#DIVFSUB
18556  074662  034577  177776          1$:     .WORD   34577,177776     ;AC
18557  074666  023400  000000          2$:     .WORD   23400.0          ;FSRC
18558  074672  051377  177776          3$:     .WORD   51377,177776     ;RES
18559  074676  000017                  4$:     17                       ;FPS BEFORE EXECUTION.
18560  074700  000000                          0                        ;FPS AFTER EXECUTION.
18561
18562
18563                                  ;DIVF TEST.
18564  074702  004737  075076          CCC8:   JSR     PC,@#DIVFSUB
18565  074706  067652  125252          1$:     .WORD   67652,125252     ;AC
18566  074712  056500  000000          2$:     .WORD   56500.0          ;FSRC
18567  074716  051343  107070          3$:     .WORD   51343,107070     ;RES
18568  074722  000000                  4$:     0                        ;FPS BEFORE EXECUTION.
18569  074724  000000                          0                        ;FPS AFTER EXECUTION.
18570
18571                                  ;DIVF WITH AC NEGATIVE, FSRC NEGATIVE.
18572  074726  004737  075076          CCC9:   JSR     PC,@#DIVFSUB
18573  074732  140400  000000          1$:     .WORD   140400.0         ;AC
18574  074736  140500  000000          2$:     .WORD   140500.0         ;FSRC
18575  074742  040052  125253          3$:     .WORD   040052,125253    ;RES
18576  074746  000000                  4$:     0                        ;FPS BEFORE EXECUTION.
18577  074750  000000                          0                        ;FPS AFTER EXECUTION.
18578
18579                                  ;DIVF WITH AC NEGATIVE AND FSRC POSITIVE.
18580  074752  004737  075076          CCC10:  JSR     PC,@#DIVFSUB
18581  074756  160077  000000          1$:     .WORD   160077,0         ;AC
18582  074762  040277  000000          2$:     .WORD   40277,0          ;FSRC
18583  074766  160000  000000          3$:     .WORD   160000.0         ;RES
18584  074772  000007                  4$:     7                        ;FPS BEFORE EXECUTION.
18585  074774  000010                          10                       ;FPS AFTER EXECUTION.
18586
18587                                  ;DIVF WITH AC POSITIVE AND FSRC NEGATIVE.
18588  074776  004737  075076          CCC11:  JSR     PC,@#DIVFSUB
18589  075002  040400  000000          1$:     .WORD   40400,0          ;AC
18590  075006  140500  000000          2$:     .WORD   140500.0         ;FSRC
18591  075012  140052  125253          3$:     .WORD   140052,125253    ;RES
18592  075016  000017                  4$:     17                       ;FPS BEFORE EXECUTION.
18593  075020  000010                          10                       ;FPS AFTER EXECUTION.
18594
18595
18596                                  ;TEST DIVF BOTH OPERANDS POSITIVE AND TRUNCATE MODE.
18597  075022  004737  075076          CCC12:  JSR     PC,@#DIVFSUB
18598  075026  060100  000001          1$:     .WORD   60100,1          ;AC
18599  075032  040300  000000          2$:     .WORD   40300,0          ;FSRC
18600  075036  060000  000000          3$:     .WORD   60000,0          ;RES
18601  075042  000052                  4$:     52                       ;FPS BEFORE EXECUTION.
18602  075044  000040                          40                       ;FPS AFTER EXECUTION.
18603
18604                                  ;DIVF WITH POSITIVE OPERANDS AND ROUND MODE.
```

```
18605  075046  004767  000024          CCC13:  JSR     PC,DIVFSUB
18606  075052  060100  000001          1$:     .WORD   60100,1         ;AC
18607  075056  040300  000000          2$:     .WORD   40300,0         ;FSRC
18608  075062  060000  000001          3$:     .WORD   60000,1         ;RES
18609  075066  000005                  4$:     5                       ;FPS BEFORE EXECUTION.
18610  075070  000000                          0                       ;FPS AFTER EXECUTION.
18611
18612  075072  000137  075214                  JMP     @#CCCDONE       ;GO TO NEXT TEST.
18613
18614                                  ;THIS SUBROUTINE, DIVFSUB, IS CALLED TO SET UP, EXECUTE
18615                                  ;AND CHECK THE RESULT OF A DIVF INSTRUCTION. IT IS CALLED THUS:
18616                                  ;
18617                                  ;               JSR     PC,@#DIVFSUB
18618                                  ;       ACARG:  .WORD   X,X             ;AC OPERAND
18619                                  ;       FSRCARG:.WORD   X,X             ;FSRC OPERAND
18620                                  ;       RES:    .WORD   X,X             ;EXPECTED RESULT
18621                                  ;       FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
18622                                  ;       FPSA:   .WORD   X               ;FPS AFTER EXECUTION
18623                                  ;       ERRES:  .WORD   X,X             ;ERROR RESULT
18624                                  ;       ERR:    ERROR   X               ;RESULT ERROR
18625                                  ;       CONT:                           ;RETURN ADDRESS
18626                                  ;
18627                                  ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC OPERAND). THEN
18628                                  ;FPSB IS LOADED INTO THE FPS. THE INSTRUCTION, DIVF IS EXECUTED.
18629                                  ;AFTER THE EXECUTION THE RESULT IS CHECKED AGAINST THE
18630                                  ;EXPECTED CORRECT RESULT, RES. IF IT IS CORRECT THEN THE FPS
18631                                  ;IS CHECKED WITH THE EXPECTED CORRECT FPS, FPSA. IF THE FPS WAS
18632                                  ;INCORRECT THEN IT IS REPORTED. IF THE RESULT WAS INCORRECT IT
18633                                  ;IS COMPARED WITH ERRES IN AN ATTEMPT TO ANALYSE THE ERROR. IF
18634                                  ;THE INCORRECT RESULT MATCHED ERRES THEN CONTROL IS PASSED TO
18635                                  ;THE ERROR CALL AT ERR. IF THE INCORRECT RESULT DID NOT MATCH ERRES
18636                                  ;THEN THE FAILURE IS REPORTED IN DIVFSUB AND CONTROL IS PASSED TO
18637                                  ;CONT. IF NO ERRORS ARE DETECTED THEN DIVFSUB RETURNS CONTROL
18638                                  ;TO CONT.
18639
18640  075076  012601                  DIVFSUB:        MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
18641  075100  012700  000200                  MOV     #200,R0         ;SET FD MODE.
18642  075104  170100                           LDFPS   R0
18643  075106  010100                           MOV     R1,R0           ;LOAD THE AC OPERAND.
18644  075110  172410                           LDD     (R0),AC0
18645  075112  016100  000014                   MOV     14(R1),R0       ;LOAD THE FPS
18646  075116  170100                           LDFPS   R0
18647  075120  010100                           MOV     R1,R0
18648  075122  062700  000004                   ADD     #4,R0           ;ESTABLISH A POINTER TO FSRC.
18649
18650  075126  174410                  1$:     DIVF    (R0),AC0        ;TEST INSTRUCTION.
18651
18652  075130  170204                           STFPS   R4              ;GET THE FPS.
18653  075132  012700  000200                   MOV     #200,R0         ;SET FD MODE
18654  075136  170100                           LDFPS   R0
18655
18656  075140  012700  075204                   MOV     #DIVFT,R0       ;GET THE RESULT OF THE DIVF.
18657  075144  174010                           STD     AC0,(R0)
18658  075146  021061  000010                   CMP     (R0),10(R1)     ;IS THE RESULT CORRECT?
18659  075152  001401                           BEQ     2$
  (2)  075154  104000                           EMT                     ;
```

```
18660  075156  026061  000002  000012  2$:     CMP     2(R0),12(R1)
18661  075164  001401                  BEQ     3$
  (2)  075166  104000                  EMT                             ;
18662  075170  026104  000016          3$:     CMP     16(R1),R4       ;IS FPS CORRECT?
18663  075174  001401                  BEQ     4$
  (2)  075176  104000                  EMT                             ;
18664  075200  000161  000020          4$:     JMP     20(R1)          ;IF NO ERRORS OCCURRED RETURN.
18665
18666  075204  000000  000000  000000  DIVFT:  .WORD   0,0,0,0
       075212  000000
18667
18668  075214                          CCCDONE:
  (1)  075214  004767  027332                  JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
18669
18670
18677
18678                                   ;*************************************************************************************
  (2)                                   ;TEST 476       DIVD TEST
  (3)                                   ;*************************************************************************************
  (2)  075220                           TS476:
18679
18680                                   ;DIVD TEST WITH POSITIVE OPERANDS AND IN ROUND MODE.
18681  075220  004737  075564          DDD1:   JSR     PC,@#DIVDSUB
18682  075224  034277  000000  000000  1$:     .WORD   34277,0,0,0     ;AC
       075232  000000
18683  075234  040277  000000  000000  2$:     .WORD   40277,0,0,0     ;FSRC
       075242  000000
18684  075244  034200  000000  000000  3$:     .WORD   34200,0,0,0     ;RES
       075252  000000
18685  075254  000200                  4$:     200                     ;FPS BEFORE EXECUTION.
18686  075256  000200                          200                     ;FPS AFTER EXECUTION.
18687
18688                                   ;DIVD WITH AC NEGATIVE AND FSRC POSITIVE IN TRUNCATE MODE.
18689  075260  004737  075564          DDD2:   JSR     PC,@#DIVDSUB
18690  075264  134277  000000  000000  1$:     .WORD   134277,0,0,0    ;AC
       075272  000000
18691  075274  040277  000000  000000  2$:     .WORD   40277,0,0,0     ;FSRC
       075302  000000
18692  075304  134200  000000  000000  3$:     .WORD   134200,0,0,0    ;RES
       075312  000000
18693  075314  000207                  4$:     207                     ;FPS BEFORE EXECUTION.
18694  075316  000210                          210                     ;FPS AFTER EXECUTION.
18695
18696                                   ;DIVD TEST WITH OPERANDS BOTH NEGATIVE AND IN TRUNCATE MODE.
18697  075320  004767  000240          DDD3:   JSR     PC,DIVDSUB
18698  075324  134300  000000  000000  1$:     .WORD   134300,0,0,1    ;AC
       075332  000001
18699  075334  140300  000000  000000  2$:     .WORD   140300,0,0,0    ;FSRC
       075342  000000
18700  075344  034200  000000  000000  3$:     .WORD   34200,0,0,0     ;RES
       075352  000000
18701  075354  000250                  4$:     250                     ;FPS BEFORE EXECUTION.
```

```
18702  075356  00C240                          240                     ;FPS AFTER EXECUTION.
18703
18704                                  ;DIVD WITH AC POSITIVE AND FSRC NEGATIVE IN ROUND MODE.
18705  075360  004737  075564          DDD4:   JSR     PC,@#DIVDSUB
18706  075364  034300  000000  000000  1$:     .WORD   34300,0,0,1     ;AC
       075372  000001
18707  075374  140300  000000  000000  2$:     .WORD   140300,0,0,0    ;FSRC
       075402  000000
18708  075404  134200  000000  000000  3$:     .WORD   134200,0,0,1    ;RES
       075412  000001
18709  075414  000207                  4$:     207                     ;FPS BEFORE EXECUTION.
18710  075416  000210                          210                     ;FPS AFTER EXECUTION.
18711
18712                                  ;DIVD TEST.
18713  075420  004737  075564          DDD5:   JSR     PC,@#DIVDSUB
18714  075424  100400  000000  000000  1$:     .WORD   100400,0,0,0    ;AC
       075432  000000
18715  075434  000500  000000  000000  2$:     .WORD   500,0,0,0       ;FSRC
       075442  000000
18716  075444  140052  125252          3$:     .WORD   140052,125252   ;RES
18717  075450  125252  125252                  .WORD   125252,125252
18718  075454  007647                  4$:     7647                    ;FPS BEFORE EXECUTION.
18719  075456  007650                          7650                    ;FPS AFTER EXECUTION.
18720
18721
18722                                  ;DIVD TEST WITH AC POSITIVE AND FSRC NEGATIVE IN ROUND MODE.
18723  075460  004737  075564          DDD6:   JSR     PC,@#DIVDSUB
18724  075464  000400  000000  000000  1$:     .WORD   400,0,0,0       ;AC
       075472  000000
18725  075474  100500  000000  000000  2$:     .WORD   100500,0,0,0    ;FSRC
       075502  000000
18726  075504  140052  125252          3$:     .WORD   140052,125252   ;RES
18727  075510  125252  125253                  .WORD   125252,125253
18728  075514  007707                  4$:     7707                    ;FPS BEFORE EXECUTION.
18729  075516  007710                          7710                    ;FPS AFTER EXECUTION.
18730
18731                                  ;DIVD TEST.
18732  075520  004737  075564          DDD7:   JSR     PC,@#DIVDSUB
18733  075524  170360  170360          1$:     .WORD   170360,170360   ;AC
18734  075530  170360  170360                  .WORD   170360,170360
18735  075534  170360  170360          2$:     .WORD   170360,170360   ;FSRC
18736  075540  170360  170360                  .WORD   170360,170360
18737  075544  040200  000000  000000  3$:     .WORD   40200,0,0,0     ;RES
       075552  000000
18738  075554  007717                  4$:     7717                    ;FPS BEFORE EXECUTION.
18739  075556  007700                          7700                    ;FPS AFTER EXECUTION.
18740
18741  075560  000137  075706                  JMP     @#DDDDONE       ;GO TO NEXT TEST.
18742
18743
18744                                  ;THIS SUBROUTINE, DIVDSUB, IS CALLED TO SET UP, EXECUTE
18745                                  ;AND CHECK THE RESULT OF A DIVD INSTRUCTION. IT IS CALLED THUS:
18746                                  ;
18747                                  ;               JSR     PC,@#DIVDSUB
18748                                  ;       ACARG:  .WORD   X,X,X,X                 ;AC OPERAND
18749                                  ;       FSRCARG:.WORD   X,X,X,X                 ;FSRC OPERAND
```

```
18750                                   ;       RES:    .WORD   X,X,X,X                 ;EXPECTED RESULT
18751                                   ;       FPSB:   .WORD   X                       ;FPS BEFORE EXECUTION
18752                                   ;       FPSA:   .WORD   X                       ;FPS AFTER EXECUTION
18753                                   ;       ERRES:  .WORD   X,X,X,X                 ;ERROR RESULT
18754                                   ;       ERR:    ERROR   X                       ;RESULT ERROR
18755                                   ;       CONT:                                   ;RETURN ADDRESS
18756                                   ;
18757                                   ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC OPERAND). THEN
18758                                   ;FPSB IS LOADED INTO THE FPS. THE INSTRUCTION, DIVD IS EXECUTED.
18759                                   ;AFTER THE EXECUTION THE RESULT IS CHECKED AGAINST THE
18760                                   ;EXPECTED CORRECT RESULT, RES. IF IT IS CORRECT THEN THE FPS
18761                                   ;IS CHECKED WITH THE EXPECTED CORRECT FPS, FPSA. IF THE FPS WAS
18762                                   ;INCORRECT THEN IT IS REPORTED. IF THE RESULT WAS INCORRECT IT
18763                                   ;IS COMPARED WITH ERRES IN AN ATTEMPT TO ANALYSE THE ERROR. IF
18764                                   ;THE INCORRECT RESULT MATCHED ERRES THEN CONTROL IS PASSED TO
18765                                   ;THE ERROR CALL AT ERR. IF THE INCORRECT RESULT DID NOT MATCH ERRES
18766                                   ;THEN THE FAILURE IS REPORTED IN DIVDSUB AND CONTROL IS PASSED TO
18767                                   ;CONT. IF NO ERRORS ARE DETECTED THEN DIVDSUB RETURNS CONTROL
18768                                   ;TO CONT.
16769
18770  075564  012601                  DIVDSUB:        MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
18771  075566  012700  000200                  MOV     #200,R0         ;SET FD MODE.
18772  075572  170100                  LDFPS   R0
18773
18774  075574  010100                  MOV     R1,R0           ;SET UP THE AC0 OPERAND.
18775  075576  172410                  LDD     (R0),AC0
18776  075600  016100  000030          MOV     30(R1),R0       ;LOAD THE FPS.
18777  075604  170100                  LDFPS   R0
18778
18779  075606  010100                  MOV     R1,R0           ;ESTABLISH A POINTER TO FSRC.
18780  075610  062700  000010          ADD     #10,R0
18781
18782  075614  174410                  1$:     DIVD    (R0),AC0        ;EXECUTE THE TEST INSTRUCTION.
18783  075616  170204                  STFPS   R4              ;GET THE FPS.
18784  075620  012700  000200          MOV     #200,R0         ;SET FD MODE.
18785  075624  170100                  LDFPS   R0
18786  075626  012700  075676          MOV     #DIVDT,R0       ;GET THE RESULT.
18787  075632  174010                  STD     AC0,(R0)
18788  075634  010102                  MOV     R1,R2           ;CHECK THE RESULT.
18789  075636  062702  000020          ADD     #20,R2
18790  075642  012703  075676          MOV     #DIVDT,R3
18791  075646  012705  000004          MOV     #4,R5
18792  075652  022223                  2$:     CMP     (R2)+,(R3)+
18793  075654  001401                  BEQ     3$
  (2)  075656  104000                  EMT                     ;
18794  075660  077504                  3$:     SOB     R5,2$
18795
18796  075662  026104  000032          CMP     32(R1),R4       ;IS FPS CORRECT?
18797  075666  001401                  BEQ     4$
  (2)  075670  104000                  EMT                     ;
18798  075672  000161  000034          4$:     JMP     34(R1)          ;RETURN.
18799  075676  000000  000000  000000  DIVDT:  .WORD   0,0,0,0
       075704  000000
18800
18801  075706                          DDDDONE:
  (1)  075706  004767  026640                  JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
```

```
  (1)                                   ;SEE IF THE USER HAS EXPRESSED
  (1)                                   ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                   ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                   ;THE USER TYPED CONTROL G?).
18802
18803
18811
18812                           ;*****************************************************************************************
  (2)                           ;TEST 477       MULF TEST
  (3)                           ;*****************************************************************************************
  (2) 075712                    TS477:
18813
18814                           ;MULF WITH (FSRC=AC=0)
18815  075712  004737  076322   EEE1:   JSR     PC,@#MULFSUB
18816  075716  000000  000000   1$:     .WORD   0,0             ;AC
18817  075722  000000  000000   2$:     .WORD   0,0             ;FSRC
18818  075726  000000  000000   3$:     .WORD   0,0             ;RES
18819  075732  007517           4$:     7517                    ;FPS BEFORE EXECUTION.
18820  075734  007504                   7504                    ;FPS AFTER EXECUTION.
18821
18822                           ;MULF WITH (FSRC=0).
18823  075736  004737  076322   EEE2:   JSR     PC,@#MULFSUB
18824  075742  071625  034435   1$:     .WORD   71625,34435     ;AC
18825  075746  000000  000000   2$:     .WORD   0,0             ;FSRC
18826  075752  000000  000000   3$:     .WORD   0,0             ;RES
18827  075756  000013           4$:     13                      ;FPS BEFORE EXECUTION.
18828  075760  000004                   4                       ;FPS AFTER EXECUTION.
18829
18830                           ;MULF WITH (AC=0)
18831  075762  004737  076322   EEE3:   JSR     PC,@#MULFSUB
18832  075766  000000  000000   1$:     .WORD   0,0             ;AC
18833  075772  071625  153443   2$:     .WORD   071625,153443   ;FSRC
18834  075776  000000  000000   3$:     .WORD   0,0             ;RES
18835  076002  007500           4$:     7500                    ;FPS BEFORE EXECUTION.
18836  076004  007504                   7504                    ;FPS AFTER EXECUTION.
18837
18838                           ;MULF WITH AC POSITIVE AND FSRC POSITIVE IN ROUND MODE.
18839  076006  004737  076322   EEE4:   JSR     PC,@#MULFSUB
18840  076012  040200  000000   1$:     .WORD   40200,0         ;AC
18841  076016  040177  177777   2$:     .WORD   40177,-1        ;FSRC
18842  076022  040177  177777   3$:     .WORD   40177,-1        ;RES
18843  076026  000017           4$:     17                      ;FPS BEFORE EXECUTION.
18844  076030  000000                   0                       ;FPS AFTER EXECUTION.
18845
18846                           ;MULF WITH AC POSITIVE AND FSRC POSITIVE IN TRUNCATE MODE.
18847  076032  004767  000264   EEE5:   JSR     PC,MULFSUB
18848  076036  040177  177777   1$:     .WORD   40177,-1        ;AC
18849  076042  040200  000000   2$:     .WORD   40200,0         ;FSRC
18850  076046  040177  177777   3$:     .WORD   40177,-1        ;RES
18851  076052  000040           4$:     40                      ;FPS BEFORE EXECUTION.
18852  076054  000040                   40                      ;FPS AFTER EXECUTION.
18853
18854                           ;MULF WITH BOTH OPERANDS POSITIVE NORMALIZE TEST.
18855  076056  004737  076322   EEE6:   JSR     PC,@#MULFSUB
18856  076062  040100  000000   1$:     .WORD   40100,0         ;AC
18857  076066  040100  000000   2$:     .WORD   40100,0         ;FSRC
```

```
18858  076072  040020  000000          3$:     .WORD   40020,0          ;RES
18859  076076  000012                  4$:     12                        ;FPS BEFORE EXECUTION.
18860  076100  000000                          0                         ;FPS AFTER EXECUTION.
18861
18862                                  ;MULF WITH BOTH OPERANDS POSITIVE IN ROUND MODE.
18863  076102  004737  076322          EEE7:   JSR     PC,@#MULFSUB
18864  076106  017500  000000          1$:     .WORD   17500,0           ;AC
18865  076112  023652  125252          2$:     .WORD   23652,125252      ;FSRC
18866  076116  003177  177777          3$:     .WORD   3177,-1           ;RES
18867  076122  007417                  4$:     7417                      ;FPS BEFORE EXECUTION.
18868  076124  007400                          7400                      ;FPS AFTER EXECUTION.
18869
18870                                  ;MULF WITH AC POSITIVE AND FSRC NEGATIVE IN ROUND MODE.
18871  076126  004737  076322          EEE8:   JSR     PC,@#MULFSUB
18872  076132  040342  000000          1$:     .WORD   40342,0           ;AC
18873  076136  176542  000000          2$:     .WORD   176542,0          ;FSRC
18874  076142  176707  102000          3$:     .WORD   176707,102000     ;RES
18875  076146  000007                  4$:     7                         ;FPS BEFORE EXECUTION.
18876  076150  000010                          10                        ;FPS AFTER EXECUTION.
18877
18878                                  ;MULF WITH AC NEGATIVE AND FSRC POSITIVE IN ROUND MODE.
18879  076152  004737  076322          EEE9:   JSR     PC,@#MULFSUB
18880  076156  140200  000000          1$:     .WORD   140200,0          ;AC
18881  076162  007417  007417          2$:     .WORD   7417,7417         ;FSRC
18882  076166  107417  007417          3$:     .WORD   107417,7417       ;RES
18883  076172  000000                  4$:     0                         ;FPS BEFORE EXECUTION.
18884  076174  000010                          10                        ;FPS AFTER EXECUTION.
18885
18886                                  ;MULF WITH BOTH OPERANDS NEGATIVE IN ROUND MODE.
18887  076176  004737  076322          EEE10:  JSR     PC,@#MULFSUB
18888  076202  144600  000000          1$:     .WORD   144600,0          ;AC
18889  076206  154000  000000          2$:     .WORD   154000,0          ;FSRC
18890  076212  060400  000000          3$:     .WORD   60400,0           ;RES
18891  076216  000017                  4$:     17                        ;FPS BEFORE EXECUTION.
18892  076220  000000                          0                         ;FPS AFTER EXECUTION.
18893
18894                                  ;MULF BOTH OPERANDS NEGATIVE IN ROUND MODE.
18895  076222  004737  076322          EEE11:  JSR     PC,@#MULFSUB
18896  076226  140300  000000          1$:     .WORD   140300,0          ;AC
18897  076232  160000  000001          2$:     .WORD   160000,1          ;FSRC
18898  076236  060100  000002          3$:     .WORD   60100,2           ;RES
18899  076242  000010                  4$:     10                        ;FPS BEFORE EXECUTION.
18900  076244  000000                          0                         ;FPS AFTER EXECUTION.
18901
18902                                  ;MULF WITH AC POSITIVE AND FSRC NEGATIVE IN TRUNCATE MODE.
18903  076246  004737  076322          EEE12:  JSR     PC,@#MULFSUB
18904  076252  060000  000001          1$:     .WORD   60000,1           ;AC
18905  076256  140300  000000          2$:     .WORD   140300,0          ;FSRC
18906  076262  160100  000001          3$:     .WORD   160100,1          ;RES
18907  076266  007547                  4$:     7547                      ;FPS BEFORE EXECUTION.
18908  076270  007550                          7550                      ;FPS AFTER EXECUTION.
18909
18910                                  ;MULF WITH AC POSITIVE AND FSRC POSITIVE IN ROUND MODE.
18911  076272  004737  076322          EEE13:  JSR     PC,@#MULFSUB
18912  076276  040277  000000          1$:     .WORD   40277,0           ;AC
18913  076302  060000  000001          2$:     .WORD   60000,1           ;FSRC
```

```
18914  076306  060077  000001          3$:     .WORD   60077,1          ;RES
18915  076312  000014                  4$:     14                      ;FPS BEFORE EXECUTION.
18916  076314  000000                          0                       ;FPS AFTER EXECUTION.
18917
18918  076316  000167  000116                  JMP     EEEDONE         ;GO TO THE NEXT TEST.
18919
18920                                  ;THIS SUBROUTINE, MULFSUB, IS CALLED TO SET UP, EXECUTE
18921                                  ;AND CHECK THE RESULT OF A MULF INSTRUCTION. IT IS CALLED THUS:
18922                                  ;
18923                                  ;               JSR     PC,@#MULFSUB
18924                                  ;       ACARG:  .WORD   X,X                     ;AC OPERAND
18925                                  ;       FSRCARG:.WORD   X,X                     ;FSRC OPERAND
18926                                  ;       RES:    .WORD   X,X                     ;EXPECTED RESULT
18927                                  ;       FPSB:   .WORD   X                       ;FPS BEFORE EXECUTION
18928                                  ;       FPSA:   .WORD   X                       ;FPS AFTER EXECUTION
18929                                  ;       ERRES:  .WORD   X,X                     ;ERROR RESULT
18930                                  ;       ERR:    ERROR   X                       ;RESULT ERROR
18931                                  ;       CONT:                                   ;RETURN ADDRESS
18932                                  ;
18933                                  ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC OPERAND). THEN
18934                                  ;FPSB IS LOADED INTO THE FPS. THE INSTRUCTION, MULF IS EXECUTED.
18935                                  ;AFTER THE EXECUTION THE RESULT IS CHECKED AGAINST THE
18936                                  ;EXPECTED CORRECT RESULT, RES. IF IT IS CORRECT THEN THE FPS
18937                                  ;IS CHECKED WITH THE EXPECTED CORRECT FPS, FPSA. IF THE FPS WAS
18938                                  ;INCORRECT THEN IT IS REPORTED. IF THE RESULT WAS INCORRECT IT
18939                                  ;IS COMPARED WITH ERRES IN AN ATTEMPT TO ANALYSE THE ERROR. IF
18940                                  ;THE INCORRECT RESULT MATCHED ERRES THEN CONTROL IS PASSED TO
18941                                  ;THE ERROR CALL AT ERR. IF THE INCORRECT RESULT DID NOT MATCH ERRES
18942                                  ;THEN THE FAILURE IS REPORTED IN MULFSUB AND CONTROL IS PASSED TO
18943                                  ;CONT. IF NO ERRORS ARE DETECTED THEN MULFSUB RETURNS CONTROL
18944                                  ;TO CONT.
18945
18946  076322  012601                  MULFSUB:        MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
18947  076324  012700  000200                  MOV     #200,R0         ;SET FD MODE.
18948  076330  170100                  LDFPS   R0
18949  076332  010100                  MOV     R1,R0           ;LOAD THE AC OPERAND.
18950  076334  172410                  LDD     (R0),AC0
18951  076336  016100  000014          MOV     14(R1),R0       ;LOAD THE FPS
18952  076342  170100                  LDFPS   R0
18953  076344  010100                  MOV     R1,R0
18954  076346  062700  000004          ADD     #4,R0           ;ESTABLISH A POINTER TO FSRC.
18955
18956  076352  171010                  1$:     MULF    (R0),AC0        ;TEST INSTRUCTION.
18957
18958  076354  170204                  STFPS   R4              ;GET THE FPS.
18959  076356  012700  000200          MOV     #200,R0         ;SET FD MODE
18960  076362  170100                  LDFPS   R0
18961
18962  076364  012700  076430          MOV     #MULFT,R0       ;GET THE RESULT OF THE MULF.
18963  076370  174010                  STD     AC0,(R0)
18964  076372  021061  000010          CMP     (R0),10(R1)     ;IS THE RESULT CORRECT?
18965  076376  001401                  BEQ     2$
  (2)  076400  104000                  EMT                     ;
18966  076402  026061  000002  000012  2$:     CMP     2(R0),12(R1)
18967  076410  001401                  BEQ     3$
  (2)  076412  104000                  EMT                     ;
```

```
18968  076414  026104  000016          3$:   CMP   16(R1),R4      ;IS FPS CORRECT?
18969  076420  001401                        BEQ   4$
  (2)  076422  104000                        EMT                  ;
18970  076424  000161  000020          4$:   JMP   20(R1)         ;IF NO ERRORS OCCURRED RETURN.
18971
18972  076430  000000  000000  000000  MULFT:.WORD 0,0,0,0
       076436  000000
18973
18974  076440                        EEEDONE:
  (1)  076440  004767  026106                JSR   PC,..RSET      ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                             ;SEE IF THE USER HAS EXPRESSED
  (1)                                                             ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                             ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                             ;THE USER TYPED CONTROL G?).
18975
18976
18984
18985                               ;*****************************************************************************
  (2)                               ;TEST 500       MULD TEST
  (3)                               ;*****************************************************************************
  (2)  076444                       TS500:
18986
18987                               ;MULD TEST WITH AC POSITIVE AND FSRC POSITIVE.
18988  076444  004737  076650        FFF1:  JSR    PC,@#MULDSUB
18989  076450  040200  000000  000000 1$:   .WORD  40200,0,0,0     ;AC
       076456  000000
18990  076460  023777  177777  177777 2$:   .WORD  23777,-1,-1,-1  ;FSRC
       076466  177777
18991  076470  023777  177777  177777 3$:   .WORD  23777,-1,-1,-1  ;RES
       076476  177777
18992  076500  000217                4$:   217                     ;FPS BEFORE EXECUTION.
18993  076502  000200                      200                     ;FPS AFTER ECECUTION.
18994
18995                               ;MULD TEST WITH BOTH OPERANDS POSITIVE TRUNCATION TEST.
18996  076504  004767  000140        FFF2:  JSR    PC,MULDSUB
18997  076510  065400  000000  000000 1$:   .WORD  65400,0,0,1     ;AC
       076516  000001
18998  076520  037577  177777  177777 2$:   .WORD  37577,-1,-1,-2  ;FSRC
       076526  177776
18999  076530  064777  177777  177777 3$:   .WORD  64777,-1,-1,-1  ;RES
       076536  177777
19000  076540  000247                4$:   247                     ;FPS BEFORE EXECUTION.
19001  076542  000240                      240                     ;FPS AFTER EXECUTION.
19002
19003                               ;MULD TEST WITH BOTH OPERANDS NEGATIVE IN ROUND MODE.
19004  076544  004737  076650        FFF3:  JSR    PC,@#MULDSUB
19005  076550  137577  177777  177777 1$:   .WORD  137577,-1,-1,-2 ;AC
       076556  177776
19006  076560  165400  000000  000000 2$:   .WORD  165400,0,0,1    ;FSRC
       076566  000001
19007  076570  065000  000000  000000 3$:   .WORD  65000,0,0,0     ;RES
       076576  000000
19008  076600  007717                4$:   7717                    ;FPS BEFORE EXECUTION.
19009  076602  007700                      7700                    ;FPS AFTER EXECUTION.
19010
19011                               ;MULD   TEST WITH AC POSITIVE AND FSRC NEGATIVE IN ROUND MODE.
```

```
19012  076604  004737  076650          FFF4:   JSR     PC,@#MULDSUB
19013  076610  017500  000000  000000  1$:     .WORD   17500,0,0,0     ;AC
       076616  000000
19014  076620  123652  125252          2$:     .WORD   123652,125252   ;FSRC
19015  076624  125252  125252                  .WORD   125252,125252
19016  076630  103177  177777  177777  3$:     .WORD   103177,-1,-1,-1 ;RES
       076636  177777
19017  076640  000200                  4$:     200                     ;FPS BEFORE EXECUTION.
19018  076642  000210                          210                     ;FPS AFTER EXECUTION.
19019
19020  076644  000167  000122                  JMP     FFFDONE
19021
19022                                  ;THIS SUBROUTINE, MULDSUB, IS CALLED TO SET UP, EXECUTE
19023                                  ;AND CHECK THE RESULT OF A MULD INSTRUCTION. IT IS CALLED THUS::
19024                                  ;               JSR     PC,@#MULDSUB
19025                                  ;       ACARG:  .WORD   X,X,X,X                 ;AC OPERAND
19026                                  ;       FSRCARG:.WORD   X,X,X,X                 ;FSRC OPERAND
19027                                  ;       RES:    .WORD   X,X,X,X                 ;EXPECTED RESULT
19028                                  ;       FPSB:   .WORD   X                       ;FPS BEFORE EXECUTION
19029                                  ;       FPSA:   .WORD   X                       ;FPS AFTER EXECUTION
19030                                  ;       ERRES:  .WORD   X,X,X,X                 ;ERROR RESULT
19031                                  ;       ERR:    ERROR   X                       ;RESULT ERROR
19032                                  ;       CONT:                                   ;RETURN ADDRESS
19033                                  ;
19034                                  ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC OPERAND). THEN
19035                                  ;FPSB IS LOADED INTO THE FPS. THE INSTRUCTION, MULD IS EXECUTED.
19036                                  ;AFTER THE EXECUTION THE RESULT IS CHECKED AGAINST THE
19037                                  ;EXPECTED CORRECT RESULT, RES. IF IT IS CORRECT THEN THE FPS
19038                                  ;IS CHECKED WITH THE EXPECTED CORRECT FPS, FPSA. IF THE FPS WAS
19039                                  ;INCORRECT THEN IT IS REPORTED. IF THE RESULT WAS INCORRECT IT
19040                                  ;IS COMPARED WITH ERRES IN AN ATTEMPT TO ANALYSE THE ERROR. IF
19041                                  ;THE INCORRECT RESULT MATCHED ERRES THEN CONTROL IS PASSED TO
19042                                  ;THE ERROR CALL AT ERR. IF THE INCORRECT RESULT DID NOT MATCH ERRES
19043                                  ;THEN THE FAILURE IS REPORTED IN MULDSUB AND CONTROL IS PASSED TO
19044                                  ;CONT. IF NO ERRORS ARE DETECTED THEN MULDSUB RETURNS CONTROL
19045                                  ;TO CONT.
19046
19047  076650  012601                  MULDSUB:        MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
19048  076652  012700  000200                  MOV     #200,R0         ;SET FD MODE.
19049  076656  170100                  LDFPS   R0
19050
19051  076660  010100                  MOV     R1,R0           ;SET UP THE AC0 OPERAND.
19052  076662  172410                  LDD     (R0),AC0
19053  076664  016100  000030          MOV     30(R1),R0       ;LOAD THE FPS.
19054  076670  170100                  LDFPS   R0
19055
19056  076672  010100                  MOV     R1,R0           ;ESTABLISH A POINTER TO FSRC.
19057  076674  062700  000010          ADD     #10,R0
19058
19059  076700  171010                  1$:     MULD    (R0),AC0        ;EXECUTE THE TEST INSTRUCTION.
19060
19061  076702  170204                  STFPS   R4              ;GET THE FPS.
19062  076704  012700  000200          MOV     #200,R0         ;SET FD MODE.
19063  076710  170100                  LDFPS   R0
19064
19065  076712  012700  076762          MOV     #MULDT,R0       ;GET THE RESULT.
```

```
19066  076716  174010                          STD     AC0,(R0)
19067  076720  010102                          MOV     R1,R2           ;CHECK THE RESULT.
19068  076722  062702  000020                  ADD     #20,R2
19069  076726  012703  076762                  MOV     #MULDT,R3
19070  076732  012705  000004                  MOV     #4,R5
19071  076736  022223                  2$:     CMP     (R2)+,(R3)+
19072  076740  001401                          BEQ     3$
  (2)  076742  104000                          EMT                     ;
19073  076744  077504                  3$:     SOB     R5,2$
19074
19075  076746  026104  000032                  CMP     32(R1),R4       ;IS FPS CORRECT?
19076  076752  001401                          BEQ     4$
  (2)  076754  104000                          EMT                     ;
19077  076756  000161  000034          4$:     JMP     34(R1)          ;RETURN.
19078
19079  076762  000000  000000  000000  MULDT:  .WORD   0,0,0,0
       076770  000000
19080  076772                          FFFDONE:
  (1)  076772  004767  025554                  JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
19081
19082
19091
19092                                   ;*****************************************************************************
  (2)                                   ;TEST 501       UNDER\OVER FLOW. USING MULF WITH TRAPS DISABLED. TEST
  (3)                                   ;*****************************************************************************
  (2)  076776                           TS501:
19093
19094                                   ;UNDERFLOW, WITH EXPONENT OF RESULT = -129
19095  076776  004737  077142          III1:   JSR     PC,@#OVUNFNT
19096  077002  020200  000000          1$:     .WORD   20200,0         ;AC
19097  077006  020000  000000          2$:     .WORD   20000,0         ;FSRC
19098  077012  000000  000000          3$:     .WORD   0,0             ;RES
19099  077016  000000                  5$:     0                       ;FPS BEFORE EXECUTION.
19100  077020  000004                          4                       ;FPS AFTER EXECUTION.
19101  077022  000012                  6$:     12                      ;FEC
19102  077024  177777                          -1                      ;FLAG
19103
19104                                   ;UNDERFLOW, WITH EXPONENT OF RESULT = -193
19105  077026  004737  077142          III2:   JSR     PC,@#OVUNFNT
19106  077032  010200  000000          1$:     .WORD   10200,0         ;AC
19107  077036  010000  000000          2$:     .WORD   10000,0         ;FSRC
19108  077042  000000  000000          3$:     .WORD   0,0             ;RES
19109  077046  005013                  5$:     5013                    ;FPS BEFORE EXECUTION.
19110  077050  005004                          5004                    ;FPS AFTER EXECUTION.
19111  077052  000012                  6$:     12                      ;FEC
19112  077054  177777                          -1                      ;FLAG
19113                                   ;OVERFLOW, EXPONENT OF RESULT = 128
19114  077056  004737  077142          III3:   JSR     PC,@#OVUNFNT
19115  077062  060200  000000          1$:     .WORD   60200,0         ;AC
19116  077066  060000  000000          2$:     .WORD   60000,0         ;FSRC
19117  077072  000000  000000          3$:     .WORD   0,0             ;RES
19118  077076  000000                  5$:     0                       ;FPS BEFORE EXECUTION.
```

```
19119  077100  000006                        6                       ;FPS AFTER EXECUTION.
19120  077102  000010              6$:       10                      ;FEC
19121  077104  000000                        0                       ;FLAG
19122                              ;OVERFLOW, EXPONENT OF RESULT = 130
19123  077106  004737  077142      III4:     JSR     PC,@#OVUNFNT
19124  077112  060200  000000      1$:       .WORD   60200,0         ;AC
19125  077116  060200  000000      2$:       .WORD   60200,0         ;FSRC
19126  077122  000000  000000      3$:       .WORD   0,0             ;RES
19127  077126  006011              5$:       6011                    ;FPS BEFORE EXECUTION.
19128  077130  006006                        6006                    ;FPS AFTER EXECUTION.
19129  077132  000010              6$:       10                      ;FEC
19130  077134  000000                        0                       ;FLAG
19131  077136  000167  000132      8$:       JMP     IIIDONE         ;GO TO NEXT TEST.
19132
19133
19134
19135                              ;THIS SUBROUTINE, OVUNFNT, IS USED TO SET UP THE OPERANDS, EXECUTE
19136                              ;THE MULF INSTRUCTION AND CHECK THE RESULTS OF AN INSTRUCTION WITH
19137                              ;OPERANDS WHICH SHOULD RESULT IN EITHER OVERFLOW OR UNDERFLOW. A CALL
19138                              ;TO IT IS MADE THUS:
19139                              ;
19140                              ;       ACARG:  .WORD   X,X             ;AC OPERAND
19141                              ;       FSRCARG:.WORD   X,X             ;FSRC OPERAND
19142                              ;       RES:    .WORD   X,X             ;EXPECTED RESULT
19143                              ;       ERRES:  .WORD   X,X             ;ERROR RESULT
19144                              ;       FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
19145                              ;       FPSA:   .WORD   X               ;FPS AFTER EXECUTION
19146                              ;       FEC:    .WORD   X               ;EXPECTED FEC
19147                              ;       FLAG:   .WORD   X               ;0/-1,OVER/UNDER FLOW FLAG
19148                              ;       ERR1:   ERROR   X               ;TRAP ERROR.
19149                              ;       BR      CONT
19150                              ;       ERR2:   ERROR   X               ;DATA, RESULT ERROR
19151                              ;       CONT:                           ;RETURN ADDRESS
19152                              ;
19153                              ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
19154                              ;THE MULF INSTRUCTION IS EXECUTED. IF NO TRAP OCCURS THEN THE
19155                              ;RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
19156                              ;COMPARED WITH FPSA IF THIS TOO IS CORRECT OVUNFNT RETURNS CONTROL
19157                              ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD OVUNFNT
19158                              ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
19159                              ;MULF IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
19160                              ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
19161                              ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN OVUNFNT
19162                              ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR2. OTHERWISE THE
19163                              ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND OVUNFNT WILL
19164                              ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.
19165                              ;IF A TRAP OCCURS (IT SHOULD NOT) THEN OVUNFNT WILL READ THE FEC.
19166                              ;SHOULD THE FEC MATCH THE ANTICIPATED FEC OVUNFNT WILL
19167                              ;STORE ALL DATA AND TRANSFER CONTROL TO THE ERROR CALL AT ERR1. IF THE
19168                              ;FEC IS NOT THE SAME AS THE ANTICIPATED FEC OVUNFNT WILL REPORT
19169                              ;THE ERROR AND RETURN TO CONT. NOTE THAT OVUNFNT USES THE FLAG
19170                              ;TO TELL WHETHER OR NOT THESE PARTICULAR OPERANDS WILL RESULT IN
19171                              ;UNDERFLOW (FLAG=-1) OR OVERFLOW (FLAG=0).
19172
19173  077142  012601              OVUNFNT:        MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
19174  077144  012700  000200              MOV     #200,R0         ;SET FD MODE.
```

```
19175  077150  170100                          LDFPS   R0
19176
19177  077152  010100                          MOV     R1,R0           ;LOAD AC0, OPERAND.
19178  077154  172410                          LDD     (R0),AC0
19179  077156  016100  000014                  MOV     14(R1),R0       ;LOAD THE FPS
19180  077162  170100                          LDFPS   R0
19181  077164  012737  077244  000244          MOV     #25$,@#FPVECT   ;SET UP THE FP TRAP VECTOR IN CASE
19182                                                                  ;OF ERROR.
19183  077172  010100                          MOV     R1,R0           ;COMPUTE THE ADDRESS OF FSRC.
19184  077174  062700  000004                  ADD     #4,R0
19185
19186  077200  171010                  1$:     MULF    (R0),AC0        ;TEST INSTRUCTION.
19187
19188  077202  170204                  2$:     STFPS   R4              ;GET FPS.
19189  077204  170305                          STST    R5              ;GET FEC.
19190  077206  012700  000200                  MOV     #200,R0         ;SET FD MODE.
19191  077212  170100                          LDFPS   R0
19192  077214  012700  077264                  MOV     #OVFNTT,R0      ;GET THE RESULT.
19193  077220  174010                          STD     AC0,(R0)
19194  077222  012700  077264                  MOV     #OVFNTT,R0      ;CHECK THE RESULT.
19195  077226  010102                          MOV     R1,R2
19196  077230  062702  000010                  ADD     #10,R2
19197  077234  012703  000002                  MOV     #2,R3
19198  077240  022022                  3$:     CMP     (R0)+,(R2)+
19199  077242  001401                          BEQ     5$
  (1)  077244                          25$:
  (2)  077244  104000                          EMT                     ;
19200  077246  077304                  5$:     SOB     R3,3$
19201
19202  077250  026104  000016                  CMP     16(R1),R4       ;WAS FPS CORRECT?
19203  077254  001401                          BEQ     4$
  (2)  077256  104000                          EMT                     ;
19204  077260  000161  000024          4$:     JMP     24(R1)          ;RETURN, TEST COMPLETED.
19205
19206  077264  000000  000000  000000  OVFNTT: .WORD   0,0,0,0
       077272  000000
19207
19208  077274                          IIIDONE:
  (1)  077274  004767  025252                  JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
19209
19210
19211
19220
19221                                   ;*************************************************************************
  (2)                                   ;TEST 502       UNDER\OVER FLOW, USING MULD WITH TRAP DISABLED, TEST
  (3)                                   ;*************************************************************************
  (2)  077300                           TS502:
19222
19223                                   ;UNDERFLOW, EXPONENT OF RESULT=-129
19224  077300  004737  077524          JJJ1:   JSR     PC,@#OVUNDNT
19225  077304  020200  000000          1$:     .WORD   20200,0         ;AC
19226  077310  127272  000000                  .WORD   127272,0
```

```
19227  077314  020000  000000  000000  2$:     .WORD   20000,0,0,0      ;FSRC
       077322  000000
19228  077324  000000  000000  000000  3$:     .WORD   0,0,0,0          ;RES
       077332  000000
19229  077334  000200                  5$:     200                      ;FPS BEFORE EXECUTION.
19230  077336  000204                          204                      ;FPS AFTER EXECUTION.
19231  077340  000012                  6$:     12                       ;FEC
19232  077342  177777                          -1                       ;FLAG
19233
19234                                  ;UNDERFLOW, EXPONENT OF RESULT = -193
19235  077344  004737  077524          JJJ2:   JSR     PC,@#OVUNDNT
19236  077350  010200  000000          1$:     .WORD   10200,0          ;AC
19237  077354  123456  000000                  .WORD   123456,0
19238  077360  010000  000000  000000  2$:     .WORD   10000,0,0,0      ;FSRC
       077366  000000
19239  077370  000000  000000  000000  3$:     .WORD   0,0,0,0          ;RES
       077376  000000
19240  077400  005213                  5$:     5213                     ;FPS BEFORE EXECUTION.
19241  077402  005204                          5204                     ;FPS AFTER EXECUTION.
19242  077404  000012                  6$:     12                       ;FEC
19243  077406  177777                          -1                       ;FLAG
19244
19245                                  ;OVERFLOW, EXPONENT OF RESULT = 128
19246  077410  004737  077524          JJJ3:   JSR     PC,@#OVUNDNT
19247  077414  060200  000000          1$:     .WORD   60200,0          ;AC
19248  077420  065432  000000                  .WORD   65432,0
19249  077424  060000  000000  000000  2$:     .WORD   60000,0,0,0      ;FSRC
       077432  000000
19250  077434  000000  000000  000000  3$:     .WORD   0,0,0,0          ;RES
       077442  000000
19251  077444  000200                  5$:     200                      ;FPS BEFORE EXECUTION.
19252  077446  000206                          206                      ;FPS AFTER EXECUTION.
19253  077450  000010                  6$:     10                       ;FEC
19254  077452  000000                          0                        ;FLAG
19255
19256                                  ;OVERFLOW, EXPONENT OF RESULT = 130
19257  077454  004737  077524          JJJ4:   JSR     PC,@#OVUNDNT
19258  077460  060200  000000          1$:     .WORD   60200,0          ;AC
19259  077464  125252  000000                  .WORD   125252,0
19260  077470  060200  000000  000000  2$:     .WORD   60200,0,0,0      ;FSRC
       077476  000000
19261  077500  000000  000000  000000  3$:     .WORD   0,0,0,0          ;RES
       077506  000000
19262  077510  006211                  5$:     6211                     ;FPS BEFORE EXECUTION.
19263  077512  006206                          6206                     ;FPS AFTER EXECUTION.
19264  077514  000010                  6$:     10                       ;FEC
19265  077516  000000                          0                        ;FLAG
19266  077520  000137  077656          8$:     JMP     @#JJJDONE        ;GO TO NEXT TEST
19267
19268                                  ;THIS SUBROUTINE, OVUNDNT, IS USED TO SET UP THE OPERANDS, EXECUTE
19269                                  ;THE MULD INSTRUCTION AND CHECK THE RESULTS OF AN INSTRUCTION WITH
19270                                  ;OPERANDS WHICH SHOULD RESULT IN EITHER OVERFLOW OR UNDERFLOW. A CALL
19271                                  ;TO IT IS MADE THUS:
19272                                  ;
19273                                  ;               ACARG:  .WORD   X,X,X,X          ;AC OPERAND
19274                                  ;               FSRCARG:.WORD   X,X,X,X          ;FSRC OPERAND
```

```
19275                                   ;               RES:    .WORD   X,X,X,X         ;EXPECTED RESULT
19276                                   ;               ERRES:  .WORD   X,X,X,X         ;ERROR RESULT
19277                                   ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
19278                                   ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
19279                                   ;               FEC:    .WORD   X               ;EXPECTED FEC
19280                                   ;               FLAG:   .WORD   X               ;0/-1,OVER/UNDER FLOW FLAG
19281                                   ;               ERR1:   ERROR   X               ;TRAP ERROR.
19282                                   ;               BR      CONT
19283                                   ;               ERR2:   ERROR   X               ;DATA, RESULT ERROR
19284                                   ;               CONT:                           ;RETURN ADDRESS
19285                                   ;
19286                                   ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
19287                                   ;THE MULD INSTRUCTION IS EXECUTED. IF NO TRAP OCCURS THEN THE
19288                                   ;RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
19289                                   ;COMPARED WITH FPSA IF THIS TOO IS CORRECT OVUNDNT RETURNS CONTROL
19290                                   ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD OVUNDNT
19291                                   ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
19292                                   ;MULD IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
19293                                   ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
19294                                   ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN OVUNDNT
19295                                   ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR2. OTHERWISE THE
19296                                   ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND OVUNDNT WILL
19297                                   ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.
19298                                   ;IF A TRAP OCCURS (IT SHOULD NOT) THEN OVUNDNT WILL READ THE FEC.
19299                                   ;SHOULD THE FEC MATCH THE ANTICIPATED FEC OVUNDNT WILL
19300                                   ;STORE ALL DATA AND TRANSFER CONTROL TO THE ERROR CALL AT ERR1. IF THE
19301                                   ;FEC IS NOT THE SAME AS THE ANTICIPATED FEC OVUNDNT WILL REPORT
19302                                   ;THE ERROR AND RETURN TO CONT. NOTE THAT OVUNDNT USES THE FLAG
19303                                   ;TO TELL WHETHER OR NOT THESE PARTICULAR OPERANDS WILL RESULT IN
19304                                   ;UNDERFLOW (FLAG=-1) OR OVERFLOW (FLAG=0).
19305
19306  077524  012601                   OVUNDNT:        MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
19307  077526  012700  000200                   MOV     #200,R0         ;SET FD MODE.
19308  077532  170100                   LDFPS   R0
19309
19310  077534  010100                   MOV     R1,R0           ;LOAD AC0, OPERAND.
19311  077536  172410                   LDD     (R0),AC0
19312  077540  016100  000030                   MOV     30(R1),R0       ;LOAD THE FPS.
19313  077544  170100                   LDFPS   R0
19314  077546  012737  077626  000244           MOV     #25$,@#FPVECT   ;SET UP THE FP TRAP VECTOR IN CASE
19315                                                                   ;OF ERROR.
19316  077554  010100                   MOV     R1,R0           ;COMPUTE THE ADDRESS OF FSRC.
19317  077556  062700  000010                   ADD     #10,R0
19318
19319  077562  171010                   1$:     MULD    (R0),AC0        ;TEST INSTRUCTION.
19320
19321  077564  170204                   2$:     STFPS   R4              ;GET FPS.
19322  077566  170305                   STST    R5              ;GET FEC.
19323  077570  012700  000200                   MOV     #200,R0         ;SET FD MODE.
19324  077574  170100                   LDFPS   R0
19325  077576  012700  077646                   MOV     #OVDNTT,R0      ;GET THE RESULT.
19326  077602  174010                   STD     AC0,(R0)
19327  077604  012700  077646                   MOV     #OVDNTT,R0      ;CHECK THE RESULT.
19328  077610  010102                   MOV     R1,R2
19329  077612  062702  000020                   ADD     #20,R2
19330  077616  012703  000004                   MOV     #4,R3
```

```
19331  077622  022022                  3$:     CMP     (R0)+,(R2)+
19332  077624  001401                          BEQ     5$
  (1)  077626                          25$:
  (2)  077626  104000                          EMT                             ;
19333  077630  077304                  5$:     SOB     R3,3$
19334
19335  077632  026104  000032                  CMP     32(R1),R4               ;WAS FPS CORRECT?
19336  077636  001401                          BEQ     4$
  (2)  077640  104000                          EMT                             ;
19337  077642  000161  000040          4$:     JMP     40(R1)                  ;RETURN, TEST COMPLETED.
19338
19339  077646  000000  000000  000000  OVDNTT: .WORD   0,0,0,0
       077654  000000
19340
19341  077656                          JJJDONE:
  (1)  077656  004767  024670                  JSR     PC,..RSET               ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                          ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                          ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                          ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                          ;THE USER TYPED CONTROL G?).
19342
19343
19344
19356
19357                                   ;*****************************************************************************
  (2)                                   ;TEST 503       UNDER\OVER FLOW, USING MULF WITH TRAPS ENABLED, TEST
  (3)                                   ;*****************************************************************************
  (2)  077662                           TS503:
19358
19359                                   ;UNDERFLOW, EXPONENT OF RESULT = -129
19360  077662  004737  100026          KKK1:   JSR     PC,@#OVUNFT
19361  077666  020123  045676          1$:     .WORD   20123,45676             ;AC
19362  077672  020200  000000          2$:     .WORD   20200,0                 ;FSRC
19363  077676  000123  045676          3$:     .WORD   123,45676               ;RES
19364  077702  002000                  5$:     2000                            ;FPS BEFORE EXECUTION.
19365  077704  102004                          102004                          ;FPS AFTER EXECUTION.
19366  077706  000012                  6$:     12                              ;FEC
19367  077710  177777                          -1                              ;FLAG
19368
19369                                   ;UNDERFLOW, EXPONENT OF THE RESULT = -193
19370  077712  004737  100026          KKK3:   JSR     PC,@#OVUNFT
19371  077716  010127  127272          1$:     .WORD   10127,127272            ;AC
19372  077722  010200  000000          2$:     .WORD   10200,0                 ;FSRC
19373  077726  060127  127272          3$:     .WORD   60127,127272            ;RES
19374  077732  007017                  5$:     7017                            ;FPS BEFORE EXECUTION.
19375  077734  107000                          107000                          ;FPS AFTER EXECUTION.
19376  077736  000012                  6$:     12                              ;FEC
19377  077740  177777                          -1
19378
19379                                   ;OVERFLOW, EXPONENT OF THE RESULT = 128
19380  077742  004737  100026          KKK4:   JSR     PC,@#OVUNFT
19381  077746  060252  125252          1$:     .WORD   60252,125252            ;AC
19382  077752  060000  000000          2$:     .WORD   60000,0                 ;FSRC
19383  077756  000052  125252          3$:     .WORD   000052,125252           ;RES
19384  077762  001000                  5$:     1000                            ;FPS BEFORE EXECUTION.
19385  077764  101006                          101006                          ;FPS AFTER EXECUTION.
```

```
19386  077766  000010                  6$:     10                      ;FEC
19387  077770  000000                          0                       ;FLAG
19388
19389                                  ;OVERFLOW, EXPONENT OF RESULT = 130
19390  077772  004737  100026          KKK5:   JSR     PC,@#OVUNFT
19391  077776  060345  067654          1$:     .WORD   60345,67654     ;AC
19392  100002  060200  000000          2$:     .WORD   60200,0         ;FSRC
19393  100006  000345  067654          3$:     .WORD   345,67654       ;RES
19394  100012  007015                  5$:     7015                    ;FPS BEFORE EXECUTION.
19395  100014  107002                          107002                  ;FPS AFTER EXECUTION.
19396  100016  000010                  6$:     10                      ;FEC
19397  100020  000000                          0                       ;FLAG
19398  100022  000167  000162          8$:     JMP     KKKDONE
19399
19400                                  ;THIS SUBROUTINE, OVUNFT, IS USED TO SET UP THE OPERANDS, EXECUTE
19401                                  ;THE MULF INSTRUCTION AND CHECK THE RESULTS OF AN INSTRUCTION WITH
19402                                  ;OPERANDS WHICH SHOULD RESULT IN EITHER OVERFLOW OR UNDERFLOW. A CALL
19403                                  ;TO IT IS MADE THUS:
19404                                  ;
19405                                  ;               ACARG:  .WORD   X,X             ;AC OPERAND
19406                                  ;               FSRCARG:.WORD   X,X             ;FSRC OPERAND
19407                                  ;               RES:    .WORD   X,X             ;EXPECTED RESULT
19408                                  ;               ERRES:  .WORD   X,X             ;ERROR RESULT
19409                                  ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
19410                                  ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
19411                                  ;               FEC:    .WORD   X               ;EXPECTED FEC
19412                                  ;               FLAG:   .WORD   X               ;0/-1,OVER/UNDER FLOW FLAG
19413                                  ;               ERR1:   ERROR   X               ;TRAP ERROR.
19414                                  ;               BR      CONT
19415                                  ;               ERR2:   ERROR   X               ;DATA, RESULT ERROR
19416                                  ;               CONT:                           ;RETURN ADDRESS
19417                                  ;
19418                                  ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
19419                                  ;THE MULF INSTRUCTION IS EXECUTED. IF THE TRAP OCCURS THEN THE
19420                                  ;RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
19421                                  ;COMPARED WITH FPSA IF THIS TOO IS CORRECT OVUNFT RETURNS CONTROL
19422                                  ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD OVUNFT
19423                                  ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. THE FEC IS TREATED
19424                                  ;IN THE SAME WAY. IF THE RESULT OF THE
19425                                  ;MULF IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
19426                                  ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
19427                                  ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN OVUNFT
19428                                  ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR2. OTHERWISE THE
19429                                  ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND OVUNFT WILL
19430                                  ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.
19431                                  ;IF NO TRAP OCCURS CONTROL IS PASSED TO ERR1.
19432                                  ;NOTE THAT OVUNFNT USES THE FLAG
19433                                  ;TO TELL WHETHER OR NOT THESE PARTICULAR OPERANDS WILL RESULT IN
19434                                  ;UNDERFLOW (FLAG=-1) OR OVERFLOW (FLAG=0).
19435
19436  100026  012601                  OVUNFT: MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
19437  100030  012700  000200                  MOV     #200,R0         ;SET FD MODE.
19438  100034  170100                          LDFPS   R0
19439  100036  010100                          MOV     R1,R0           ;LOAD AC0, OPERAND.
19440  100040  172410                          LDD     (R0),AC0
19441  100042  016100  000014                  MOV     14(R1),R0       ;LOAD THE FPS.
```

```
19442  100046  170100                          LDFPS   R0
19443  100050  012737  100072  0C0244          MOV     #50$,@#FPVECT   ;SET UP THE FP TRAP VECTOR IN CASE
19444                                                                  ;OF ERROR.
19445  100056  010100                          MOV     R1,R0           ;COMPUTE THE ADDRESS OF FSRC.
19446  100060  062700  000004                  ADD     #4,R0
19447
19448  100064  171010                  1$:     MULF    (R0),AC0        ;TEST INSTRUCTION. SHOULD CAUSE TRAP.
19449  100066  170000                  2$:     CFCC
19450  100070  104000                          EMT                     ;
19451  100072  011602                  50$:    MOV     (SP),R2         ;TRAP TO HERE AND SEE IF THE PC OF THE
19452  100074  020227  100066                  CMP     R2,#2$          ;TRAP WAS THAT OF THE MULF INSTRUCTION.
19453  100100  001401                          BEQ     51$
  (2)  100102  104000                          EMT                     ;
19454  100104  022626                  51$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
19455  100106  170204                          STFPS   R4              ;GET FPS.
19456  100110  170305                          STST    R5              ;GET FEC.
19457  100112  012700  000200                  MOV     #200,R0         ;SET FD MODE.
19458  100116  170100                          LDFPS   R0
19459  100120  012700  100200                  MOV     #OVFTT,R0       ;GET THE RESULT.
19460  100124  174010                          STD     AC0,(R0)
19461  100126  012700  100200                  MOV     #OVFTT,R0       ;CHECK THE RESULT.
19462  100132  010102                          MOV     R1,R2
19463  100134  062702  000010                  ADD     #10,R2
19464  100140  012703  000002                  MOV     #2,R3
19465  100144  022022                  3$:     CMP     (R0)+,(R2)+
19466  100146  001401                          BEQ     5$
  (2)  100150  104000                          EMT                     ;
19467  100152  077304                  5$:     SOB     R3,3$
19468
19469  100154  026104  000016                  CMP     16(R1),R4       ;WAS FPS CORRECT?
19470  100160  001401                          BEQ     6$
  (2)  100162  104000                          EMT                     ;
19471  100164  026105  000020          6$:     CMP     20(R1),R5       ;IS FEC CORRECT?
19472  100170  001401                          BEQ     4$
  (2)  100172  104000                          EMT                     ;
19473  100174  000161  000024          4$:     JMP     24(R1)          ;RETURN, TEST COMPLETED.
19474
19475  100200  000000  000000  000000  OVFTT:  .WORD   0,0,0,0
       100206  000000
19476
19477  100210                          KKKDONE:
  (1)  100210  004767  024336                  JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
19478
19479
19480
19488
19489                           ;*************************************************************************
  (2)                           ;TEST 504       UNDER\OVER FLOW, USING MULD WITH TRAPS ENABLED, TEST
  (3)                           ;*************************************************************************
  (2)  100214                   TS504:
19490
19491                           ;UNDERFLOW, EXPONENT OF RESULT = -129
```

```
19492  100214  004737  100440          LLL1:   JSR     PC,@#OVUNDT
19493  100220  020052  125252          1$:     .WORD   20052,125252    ;AC
19494  100224  125252  125252                  .WORD   125252,125252
19495  100230  020300  000000  000000  2$:     .WORD   20300,0,0,0     ;FSRC
       100236  000000
19496  100240  000177  177777  177777  3$:     .WORD   177,-1,-1,-1    ;RES
       100246  177777
19497  100250  002200                  5$:     2200                    ;FPS BEFORE EXECUTION.
19498  100252  102204                          102204                  ;FPS AFTER EXECUTION.
19499  100254  000012                  6$:     12                      ;FEC
19500  100256  177777                          -1                      ;FLAG
19501
19502                                  ;UNDERFLOW, EXPONENT OF THE RESULT = -193
19503  100260  004737  100440          LLL2:   JSR     PC,@#OVUNDT
19504  100264  010327  127272          1$:     .WORD   10327,127272    ;AC
19505  100270  036363  045454                  .WORD   36363,45454
19506  100274  010000  000000  000000  2$:     .WORD   10000,0,0,0     ;FSRC
       100302  000000
19507  100304  060127  127272          3$:     .WORD   60127,127272    ;RES
19508  100310  036363  045454                  .WORD   36363,45454
19509  100314  007217                  5$:     7217                    ;FPS BEFORE EXECUTION.
19510  100316  107200                          107200                  ;FPS AFTER EXECUTION.
19511  100320  000012                  6$:     12                      ;FEC
19512  100322  177777                          -1                      ;FLAG
19513
19514                                  ;OVERFLOW, EXPONENT OF THE RESULT = 128
19515  100324  004737  100440          LLL3:   JSR     PC,@#OVUNDT
19516  100330  060252  125252          1$:     .WORD   60252,125252    ;AC
19517  100334  125252  125252                  .WORD   125252,125252   ;FSRC
19518  100340  160100  000000  000000  2$:     .WORD   160100,0,0,0    ;FSRC
       100346  000000
19519  100350  100177  177777  177777  3$:     .WORD   100177,-1,-1,-1 ;RES
       100356  177777
19520  100360  001200                  5$:     1200                    ;FPS BEFORE EXECUTION.
19521  100362  101216                          101216                  ;FPS AFTER EXECUTION.
19522  100364  000010                  6$:     10                      ;FEC
19523  100366  000000                          0                       ;FLAG
19524
19525                                  ;OVERFLOW, EXPONENT OF THE RESULT = 130
19526  100370  004737  100440          LLL4:   JSR     PC,@#OVUNDT
19527  100374  060345  067654          1$:     .WORD   60345,67654     ;AC
19528  100400  056765  045676                  .WORD   56765,45676
19529  100404  060200  000000  000000  2$:     .WORD   60200,0,0,0     ;FSRC
       100412  000000
19530  100414  000345  067654          3$:     .WORD   345,67654       ;RES
19531  100420  056765  045676                  .WORD   56765,45676
19532  100424  007215                  5$:     7215                    ;FPS BEFORE EXECUTION.
19533  100426  107202                          107202                  ;FPS AFTER EXECUTION.
19534  100430  000010                  6$:     10                      ;FEC
19535  100432  000000                          0                       ;FLAG
19536  100434  000137  100622          8$:     JMP     @#LLLDONE
19537
19538                                  ;THIS SUBROUTINE, OVUNDT, IS USED TO SET UP THE OPERANDS, EXECUTE
19539                                  ;THE MULD INSTRUCTION AND CHECK THE RESULTS OF AN INSTRUCTION WITH
19540                                  ;OPERANDS WHICH SHOULD RESULT IN EITHER OVERFLOW OR UNDERFLOW. A CALL
19541                                  ;TO IT IS MADE THUS:
```

```
19542                                   ;
19543                                   ;               ACARG:  .WORD   X,X,X,X         ;AC OPERAND
19544                                   ;               FSRCARG:.WORD   X,X,X,X         ;FSRC OPERAND
19545                                   ;               RES:    .WORD   X,X,X,X         ;EXPECTED RESULT
19546                                   ;               ERRES:  .WORD   X,X,X,X         ;ERROR RESULT
19547                                   ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
```

```
19549                                   ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
19550                                   ;               FEC:    .WORD   X               ;EXPECTED FEC
19551                                   ;               FLAG:   .WORD   X               ;0/-1,OVER/UNDER FLOW FLAG
19552                                   ;               ERR1:   ERROR   X               ;TRAP ERROR.
19553                                   ;               BR      CONT
19554                                   ;               ERR2:   ERROR   X               ;DATA, RESULT ERROR
19555                                   ;               CONT:                           ;RETURN ADDRESS
19556                                   ;
19557                                   ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
19558                                   ;THE MULD INSTRUCTION IS EXECUTED. IF THE TRAP OCCURS THEN THE
19559                                   ;RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
19560                                   ;COMPARED WITH FPSA IF THIS TOO IS CORRECT OVUNDT RETURNS CONTROL
19561                                   ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD OVUNDT
19562                                   ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. THE FEC IS TREATED
19563                                   ;IN THE SAME WAY. IF THE RESULT OF THE
19564                                   ;MULF IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
19565                                   ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
19566                                   ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN OVUNDT
19567                                   ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR2. OTHERWISE THE
19568                                   ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND OVUNDT WILL
19569                                   ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.
19570                                   ;IF NO TRAP OCCURS CONTROL IS PASSED TO ERR1.
19571                                   ;NOTE THAT OVUNDNT USES THE FLAG
19572                                   ;TO TELL WHETHER OR NOT THESE PARTICULAR OPERANDS WILL RESULT IN
19573                                   ;UNDERFLOW (FLAG=-1) OR OVERFLOW (FLAG=0).
19574
19575  100440  012601                   OVUNDT: MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
19576  100442  012700  000200                   MOV     #200,R0         ;SET FD MODE.
19577  100446  170100                           LDFPS   R0
19578
19579  100450  010100                           MOV     R1,R0           ;LOAD AC0, OPERAND.
19580  100452  172410                           LDD     (R0),AC0
19581  100454  016100  000030                   MOV     30(R1),R0       ;LOAD THE FPS.
19582  100460  170100                           LDFPS   R0
19583  100462  012737  100504  000244           MOV     #50$,@#FPVECT   ;SET UP THE FP TRAP VECTOR IN CASE
19584                                                                   ;OF ERROR.
19585  100470  010100                           MOV     R1,R0           ;COMPUTE THE ADDRESS OF FSRC.
19586  100472  062700  000010                   ADD     #10,R0
19587
19588  100476  171010                   1$:     MULD    (R0),AC0        ;TEST INSTRUCTION. SHOULD CAUSE TRAP.
19589  100500  170000                   2$:     CFCC
19590  100502  104000                           EMT                     ;
19591  100504  011602                   50$:    MOV     (SP),R2         ;TRAP TO HERE AND SEE IF THE PC OF THE
19592  100506  020227  100500                   CMP     R2,#2$          ;TRAP WAS THAT OF THE MULF INSTRUCTION.
19593  100512  001401                           BEQ     51$             ;BRANCH IF YES.
19594  100514  104000                           EMT                     ;
19595  100516  022626                   51$:    CMP     (SP)+,(SP)+     ;RESET THE STACK
19596  100520  170204                           STFPS   R4              ;GET FPS.
19597  100522  170305                           STST    R5              ;GET FEC.
19598  100524  012700  000200                   MOV     #200,R0         ;SET FD MODE.
19599  100530  170100                           LDFPS   R0
19600  100532  012700  100612                   MOV     #OVDTT,R0       ;GET THE RESULT.
19601  100536  174010                           STD     AC0,(R0)
19602  100540  012700  100612                   MOV     #OVDTT,R0       ;CHECK THE RESULT.
19603  100544  010102                           MOV     R1,R2
19604  100546  062702  000020                   ADD     #20,R2
```

```
19605  100552  012703  000004          MOV     #4,R3
19606  100556  022022          3$:     CMP     (R0)+,(R2)+
19607  100560  001401                  BEQ     5$
  (2)  100562  104000                  EMT                     ;
19608  100564  077304          5$:     SOB     R3,3$
19609  100566  026104  000032          CMP     32(R1),R4       ;WAS FPS CORRECT?
19610  100572  001401                  BEQ     6$
  (2)  100574  104000                  EMT                     ;
19611  100576  026105  000034          6$:     CMP     34(R1),R5       ;IS FEC CORRECT?
19612  100602  001401                  BEQ     4$
  (2)  100604  104000                  EMT                     ;
19613  100606  000161  000040          4$:     JMP     40(R1)          ;RETURN, TEST COMPLETED.
19614
19615  100612  000000  000000  000000  OVDTT:  .WORD   0,0,0,0
       100620  000000
19616
19617  100622                  LLLDONE:
  (1)  100622  004767  023724          JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                         ;SEE IF THE USER HAS EXPRESSED
  (1)                                                         ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                         ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                         ;THE USER TYPED CONTROL G?).
19618
19619
19620
19621
19629
19630                   ;*****************************************************************************************
  (2)                   ;TEST 505       MODF TEST
  (3)                   ;*****************************************************************************************
  (2)  100626           TS505:
19631
19632                   ;MODF WITH (FSRC=AC=0)
19633  100626  004737  101352  GGG1:   JSR     PC,@#MODFSUB
19634  100632  000000  000000  1$:     .WORD   0,0             ;AC
19635  100636  000000  000000  2$:     .WORD   0,0             ;FSRC
19636  100642  000000  000000  3$:     .WORD   0,0             ;FRACTIONAL RES.
19637  100646  000000  000000  4$:     .WORD   0,0             ;INTEGER RES.
19638  100652  000013          7$:     13                      ;FPS BEFORE EXECUTION.
19639  100654  000004                  4                       ;FPS AFTER EXECUTION.
19640
19641                   ;MODF TEST, WITH (FSRC=0)
19642  100656  004737  101352  GGG2:   JSR     PC,@#MODFSUB
19643  100662  123456  076543  1$:     .WORD   123456,76543    ;AC
19644  100666  000000  000000  2$:     .WORD   0,0             ;FSRC
19645  100672  000000  000000  3$:     .WORD   0,0             ;FRACTIONAL RES.
19646  100676  000000  000000  4$:     .WORD   0,0             ;INTEGER RESULT.
19647  100702  000000          7$:     0                       ;FPS BEFORE EXECUTION.
19648  100704  000004                  4                       ;FPS AFTER EXECUTION.
19649
19650                   ;MODF TEST WITH (AC=0)
19651  100706  004737  101352  GGG3:   JSR     PC,@#MODFSUB
19652  100712  000000  000000  1$:     .WORD   0,0             ;AC
19653  100716  076543  021234  2$:     .WORD   76543,21234     ;FSRC
19654  100722  000000  000000  3$:     .WORD   0,0             ;FRACTIONAL RES.
19655  100726  000000  000000  4$:     .WORD   0,0             ;INTEGER RES.
```

```
19656  100732  000003          7$:   3                    ;FPS BEFORE EXECUTION.
19657  100734  000004                4                    ;FPS AFTER EXECUTION.
19658
19659                          ;MODF TEST WITH EXPONENT OF THE RESULT = 25
19660  100736  004737  101352  GGG4:  JSR    PC,@#MODFSUB
19661  100742  046252  125252  1$:    .WORD  46252,125252  ;AC
19662  100746  040300  000000  2$:    .WORD  40300,0       ;FSRC
19663  100752  000000  000000  3$:    .WORD  0,0           ;FRACTIONAL RES.
19664  100756  046377  177777  4$:    .WORD  46377,-1      ;INTEGER RES.
19665  100762  000013          7$:   13                   ;FPS BEFORE EXECUTION.
19666  100764  000004                4                    ;FPS AFTER EXECUTION.
19667
19668                          ;MODF TEST WITH EXPONENT OF THE RESULT = 127
19669  100766  004737  101352  GGG5:  JSR    PC,@#MODFSUB
19670  100772  077652  125252  1$:    .WORD  77652,125252  ;AC
19671  100776  040300  000000  2$:    .WORD  40300,0       ;FSRC
19672  101002  000000  000000  3$:    .WORD  0,0           ;FRACTIONAL RES.
19673  101006  077777  177777  4$:    .WORD  77777,-1      ;INTEGER RES.
19674  101012  000000          7$:   0                    ;FPS BEFORE EXECUTION.
19675  101014  000004                4                    ;FPS AFTER EXECUTION.
19676
19677                          ;MODF TEST WITH EXPONENT OF RESULT = 25
19678  101016  004737  101352  GGG6:  JSR    PC,@#MODFSUB
19679  101022  046200  000001  1$:    .WORD  46200,1       ;AC
19680  101026  040340  000000  2$:    .WORD  40340,0       ;FSRC
19681  101032  000000  000000  3$:    .WORD  0,0           ;FRACTIONAL RES.
19682  101036  046340  000001  4$:    .WORD  46340,1       ;INTEGER RES.
19683  101042  000013          7$:   13                   ;FPS BEFORE EXECUTION.
19684  101044  000004                4                    ;FPS AFTER EXECUTION.
19685
19686                          ;MODF  TEST WITH EXPONENT OF THE RESULT = 24
19687  101046  004737  101352  GGG7:  JSR    PC,@#MODFSUB
19688  101052  046000  000001  1$:    .WORD  46000,1       ;AC
19689  101056  040340  000000  2$:    .WORD  40340,0       ;FSRC
19690  101062  040100  000000  3$:    .WORD  40100,0       ;FRACTIONAL RES.
19691  101066  046140  000001  4$:    .WORD  46140,1       ;INTEGER RESULT.
19692  101072  000000          7$:   0                    ;FPS BEFORE EXECUTION.
19693  101074  000000                0                    ;FPS AFTER EXECUTION.
19694
19695                          ;MODF TEST WITH EXPONENT OF THE RESULT = 10
19696  101076  004737  101352  GGG8:  JSR    PC,@#MODFSUB
19697  101102  042577  177777  1$:    .WORD  42577,-1      ;AC
19698  101106  040200  000000  2$:    .WORD  40200,0       ;FSRC
19699  101112  040177  176000  3$:    .WORD  40177,176000  ;FRACTIONAL RES.
19700  101116  042577  140000  4$:    .WORD  42577,140000  ;INTEGER RES.
19701  101122  000000          7$:   0                    ;FPS BEFORE EXECUTION.
19702  101124  000000                0                    ;FPS AFTER EXECUTION.
19703
19704                          ;MODF TEST WITH THE EXPONENT OF THE RESULT = 10
19705  101126  004737  101352  GGG9:  JSR    PC,@#MODFSUB
19706  101132  042577  140001  1$:    .WORD  42577,140001  ;AC
19707  101136  040200  000000  2$:    .WORD  40200,0       ;FSRC
19708  101142  034600  000000  3$:    .WORD  34600,0       ;FRACTIONAL RES.
19709  101146  042577  140000  4$:    .WORD  42577,140000  ;INTEGER RES.
19710  101152  000000          7$:   0                    ;FPS BEFORE EXECUTION.
19711  101154  000000                0                    ;FPS AFTER EXECUTION.
```

```
19712
19713                                   ;MODF TEST WITH EXPONENT OF THE RESULT = 9
19714  101156  004737  101352           GGG10:  JSR     PC,@#MODFSUB
19715  101162  042377  100000           1$:     .WORD   42377,100000    ;AC
19716  101166  040200  000000           2$:     .WORD   40200,0         ;FSRC
19717  101172  000000  000000           3$:     .WORD   0,0             ;FRACTIONAL RES.
19718  101176  042377  100000           4$:     .WORD   42377,100000    ;INTEGER RES.
19719  101202  000013                   7$:     13                      ;FPS BEFORE EXECUTION.
19720  101204  000004                           4                       ;FPS AFTER EXECUTION.
19721
19722                                   ;MODF TEST WITH EXPONENT OF THE RESULT = 0
19723  101206  004737  101352           GGG11:  JSR     PC,@#MODFSUB
19724  101212  040177  177777           1$:     .WORD   40177,-1        ;AC
19725  101216  040200  000000           2$:     .WORD   40200,0         ;FSRC
19726  101222  040177  177777           3$:     .WORD   40177,-1        ;FRACTIONAL RES.
19727  101226  000000  000000           4$:     .WORD   0,0             ;INTEGER RES.
19728  101232  000017                   7$:     17                      ;FPS BEFORE EXECUTION.
19729  101234  000000                           0                       ;FPS AFTER EXECUTION.
19730
19731                                   ;MODF TEST WITH EXPONENT OF THE RESULT = -15
19732  101236  004737  101352           GGG12:  JSR     PC,@#MODFSUB
19733  101242  034377  177777           1$:     .WORD   34377,-1        ;AC
19734  101246  040200  000000           2$:     .WORD   40200,0         ;FSRC
19735  101252  034377  177777           3$:     .WORD   34377,-1        ;FRACTIONAL RES.
19736  101256  000000  000000           4$:     .WORD   0,0             ;INTEGER RES.
19737  101262  000000                   7$:     0                       ;FPS BEFORE EXECUTION.
19738  101264  000000                           0                       ;FPS AFTER EXECUTION.
19739
19740                                   ;MODF TEST WITH EXPONENT OF RESULT = -64, IN ROUND MODE
19741  101266  004737  101352           GGG13:  JSR     PC,@#MODFSUB
19742  101272  020000  000001           1$:     .WORD   20000,1         ;AC
19743  101276  040300  000000           2$:     .WORD   40300,0         ;FSRC
19744  101302  020100  000002           3$:     .WORD   20100,2         ;FRACTIONAL RES.
19745  101306  000000  000000           4$:     .WORD   0,0             ;INTEGER RES.
19746  101312  000000                   7$:     0                       ;FPS BEFORE EXECUTION.
19747  101314  000000                           0                       ;FPS AFTER EXECUTION.
19748
19749                                   ;MODF TEST WITH EXPONENT OF RESULT = 11
19750  101316  004737  101352           GGG14:  JSR     PC,@#MODFSUB
19751  101322  142777  170000           1$:     .WORD   142777,170000   ;AC
19752  101326  040200  000000           2$:     .WORD   40200,0         ;FSRC
19753  101332  140000  000000           3$:     .WORD   140000,0        ;FRACTIONAL RES.
19754  101336  142777  160000           4$:     .WORD   142777,160000   ;INTEGER RES.
19755  101342  000007                   7$:     7                       ;FPS BEFORE EXECUTION.
19756  101344  000010                           10                      ;FPS AFTER EXECUTION.
19757  101346  000167  000204           9$:     JMP     GGGDONE         ;GO TO NEXT TEST.
19758
19759
19760                                   ;THIS SUBROUTINE, MODFSUB, IS CALLED TO SETUP THE
19761                                   ;OPERANDS, EXECUTE THE MODF INSTRUCTION AND CHECK THE RESULTS.
19762                                   ;IT IS CALLED THUS:
19763                                   ;
19764                                   ;               ACARG:  .WORD   X,X             ;AC OPERAND
19765                                   ;               FSRCARG:.WORD   X,X             ;FSRC OPERAND
19766                                   ;               FRES:   .WORD   X,X             ;FRACTIONAL RESULT
19767                                   ;               INTRES: .WORD   X,X             ;INTEGER RESULT
```

```
19768                                   ;              ERFRES: .WORD   X,X             ;ERROR FRACTION RESULT
19769                                   ;              ERINTRES:.WORD  X,X             ;ERROR INTEGER RESULT
19770                                   ;              FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
19771                                   ;              FPSA:   .WORD   X               ;FPS AFTER EXECUTION
19772                                   ;              ERR1:   ERROR   X               ;FRACTION ERROR
19773                                   ;                      BR      CONT
19774                                   ;              ERR2:   ERROR   X               ;INTEGER ERROR
19775                                   ;              CONT:                           ;RETURN ADDRESS
19776                                   ;
19777                                   ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC ARGUMENT). THE MODF
19778                                   ;INSTRUCTION IS EXECUTED. THEN THE RESULTS ARE RETRIEVED.
19779                                   ;THE FRACTION PART OF THE RESULT IS COMPARED WITH FRES. IF THIS IS CORRECT
19780                                   ;THEN THE INTEGER PART IS COMPARED WITH INTRES. IF BOTH OF THESE ARE CORRECT
19781                                   ;THEN THE FPS IS COMPARED WITH FPSA. AFTER EXECUTION IF NO ERRORS OCCURRED
19782                                   ;THEN MODFSUB WILL RETURN TO CONT. IF THE FPS WAS INCORRECT
19783                                   ;IT IS REPORTED HERE. IF THE FRACTION IS INCORRECT IT IS COMPARED WITH
19784                                   ;THE ANTICIPATED BAD FRACTION, ERFRES. IF THIS DUESN'T MATCH
19785                                   ;THE TRUE RESULT THEN THE ERROR IS REPORTED HERE. IF THE ANTICIPATED
19786                                   ;FAILURE MATCHES THE TRUE RESULT THEN  MODFSUB PASSES CONTROL TO THE
19787                                   ;ERROR CALL AT ERR1. LIKEWISE IF THE INTEGER PART OF THE RESULT IS
19788                                   ;NOT CORRECT THEN IT IS COMPARED WITH THE ANTICIPATED INTEGER
19789                                   ;FAILURE. IF THIS DOEN'T MATCH THEN THE ERROR IS REPORTED HERE.
19790                                   ;IF A MATCH IS MADE HOWEVER, MODFSUB WILL RETURN CONTROL TO THE ERROR
19791                                   ;CALL AT ERR2.
19792
19793  101352  012601                   MODFSUB:        MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS
19794  101354  012700  000200                   MOV     #200,R0         ;SET FD MODE.
19795  101360  170100                   LDFPS   R0
19796  101362  010100                   MOV     R1,R0           ;SET UP AC0
19797  101364  172410                   LDD     (R0),AC0
19798  101366  012700  101546           MOV     #MODP1,R0       ;PUT A BACKROUND PATTERN INTO AC1.
19799  101372  172510                   LDD     (R0),AC1
19800  101374  016100  000020           MOV     20(R1),R0       ;SET UP THE FPS.
19801  101400  170100                   LDFPS   R0
19802  101402  010100                   MOV     R1,R0           ;COMPUTE THE ADDRESS OF THE FSRC.
19803  101404  062700  000004           ADD     #4,R0
19804
19805  101410  171410           1$:     MODF    (R0),AC0        ;EXECUTE THE TEST INSTRUCTION.
19806
19807  101412  170204                   STFPS   R4              ;GET THE FPS.
19808  101414  012700  000200           MOV     #200,R0         ;SET FD MODE.
19809  101420  170100                   LDFPS   R0
19810  101422  012700  101526           MOV     #MODFT0,R0      ;GET THE FRACTIONAL RESULT.
19811  101426  174010                   STD     AC0,(R0)
19812  101430  012700  101536           MOV     #MODFT1,R0      ;GET THE INTEGER RESULT.
19813  101434  174110                   STD     AC1,(R0)
19814  101436  012702  101526           MOV     #MODFT0,R2      ;CHECK THE FRACTIONAL RESULT.
19815  101442  026112  000010           CMP     10(R1),(R2)
19816  101446  001401                   BEQ     2$
  (2)  101450  104000                   EMT                     ;
19817  101452  026162  000012  000002  2$:     CMP     12(R1),2(R2)
19818  101460  001401                   BEQ     3$
  (2)  101462  104000                   EMT                     ;
19819  101464  012702  101536           3$:     MOV     #MODFT1,R2      ;CHECK THE INTEGER RESULT.
19820  101470  026112  000014           CMP     14(R1),(R2)
19821  101474  001401                   BEQ     4$
```

```
  (2)  101476  104000                          EMT                     ;
19822  101500  026162  000016  000002  4$:     CMP     16(R1),2(R2)
19823  101506  001401                          BEQ     5$
  (2)  101510  104000                          EMT                     ;
19824  101512  026104  000022          5$:     CMP     22(R1),R4       ;CHECK THE FPS.
19825  101516  001401                          BEQ     9$
  (2)  101520  104000                          EMT                     ;
19826  101522  000161  000024          9$:     JMP     24(R1)          ;RETURN.
19827
19828  101526  000000  000000  000000  MODFT0: .WORD   0,0,0,0
       101534  000000
19829
19830  101536  000000  000000  000000  MODFT1: .WORD   0,0,0,0
       101544  000000
19831
19832  101546  177777  177777  177777  MODP1:  .WORD   -1,-1,-1,-1
       101554  177777
19833
19834  101556                          GGGDONE:
  (1)  101556  004767  022770                  JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
19835
19836
19837
19845
19846                                  ;*******************************************************************************
  (2)                                  ;TEST 506       MODD TEST
  (3)                                  ;*******************************************************************************
  (2)  101562                          TS506:
19847
19848                                  ;MODD WITH (FSRC=AC=0)
19849  101562  004737  102576          HHH1:   JSR     PC,@#MODDSUB
19850  101566  000000  000000  000000  1$:     .WORD   0,0,0,0         ;AC
       101574  000000
19851  101576  000000  000000  000000  2$:     .WORD   0,0,0,0         ;FSRC
       101604  000000
19852  101606  000000  000000  000000  3$:     .WORD   0,0,0,0         ;FRACTIONAL RES.
       101614  000000
19853  101616  000000  000000  000000  4$:     .WORD   0,0,0,0         ;INTEGER RES.
       101624  000000
19854  101626  000200                  .$:     200                     ;FPS BEFORE EXECUTION.
19855  101630  000204                          204                     ;FPS AFTER EXECUTION.
19856
19857                                  ;MODD TEST WITH FSRC=0
19858  101632  004737  102576          HHH2:   JSR     PC,@#MODDSUB
19859  101636  012345  067012          1$:     .WORD   012345,67012    ;AC
19860  101642  034567  012345                  .WORD   34567,012345
19861  101646  000000  000000  000000  2$:     .WORD   0,0,0,0         ;FSRC
       101654  000000
19862  101656  000000  000000  000000  3$:     .WORD   0,0,0,0         ;FRACTIONAL RES.
       101664  000000
19863  101666  000000  000000  000000  4$:     .WORD   0,0,0,0         ;INTEGER RES.
       101674  000000
```

```
19864  101676  000213                    7$:     213                     ;FPS BEFORE EXECUTION.
19865  101700  000204                            204                     ;FPS AFTER EXECUTION.
19866
19867                                    ;MODD TEST WITH (AC=0)
19868  101702  004737  102576            MMM3:   JSR     PC,@#MODDSUB
19869  101706  000000  000000  000000    1$:     .WORD   0,0,0,0         ;AC
       101714  000000
19870  101716  072727  127272            2$:     .WORD   72727,127272    ;FSRC
19871  101722  072727  127272                    .WORD   72727,127272
19872  101726  000000  000000  000000    3$:     .WORD   0,0,0,0         ;FRACTIONAL RES.
       101734  000000
19873  101736  000000  000000  000000    4$:     .WORD   0,0,0,0         ;INTEGER RES.
       101744  000000
19874  101746  000213                    7$:     213                     ;FPS BEFORE EXECUTION.
19875  101750  000204                            204                     ;FPS AFTER EXECUTION.
19876
19877                                    ;MODD TEST WITH EXPONENT OF THE RESULT = 57
19878  101752  004737  102576            MMM4:   JSR     PC,@#MODDSUB
19879  101756  056252  125252            1$:     .WORD   56252,125252    ;AC
19880  101762  125252  125250                    .WORD   125252,125250
19881  101766  040300  000000  000000    2$:     .WORD   40300,0,0,0     ;FSRC
       101774  000000
19882  101776  000000  000000  000000    3$:     .WORD   0,0,0,0         ;FRACTIONAL RES.
       102004  000000
19883  102006  056377  177777  177777    4$:     .WORD   56377,-1,-1,-4  ;INTEGER RES.
       102014  177774
19884  102016  000213                    7$:     213                     ;FPS BEFORE EXECUTION.
19885  102020  000204                            204                     ;FPS AFTER EXECUTION.
19886
19887                                    ;MODD TEST WITH EXPONENT OF THE RESULT = 79
19888  102022  004737  102576            MMM5:   JSR     PC,@#MODDSUB
19889  102026  140240  000000  000000    1$:     .WORD   140240,0,0,0    ;AC
       102034  000000
19890  102036  063714  146314            2$:     .WORD   63714,146314    ;FSRC
19891  102042  133572  167737                    .WORD   133572,167737
19892  102046  000000  000000  000000    3$:     .WORD   0,0,0,0         ;FRACTIONAL RES.
       102054  000000
19893  102056  163777  177777            4$:     .WORD   163777,-1       ;INTEGER RES.
19894  102062  162531  125726                    .WORD   162531,125726
19895  102066  000210                    7$:     210                     ;FPS BEFORE EXECUTION.
19896  102070  000204                            204                     ;FPS AFTER EXECUTION.
19897
19898                                    ;MODD TEST WITH EXPONENT OF THE RESULT = 57
19899  102072  004737  102576            MMM6:   JSR     PC,@#MODDSUB
19900  102076  056200  000000  000000    1$:     .WORD   56200,0,0,1     ;AC
       102104  000001
19901  102106  040340  000000  000000    2$:     .WORD   40340,0,0,0     ;FSRC
       102114  000000
19902  102116  000000  000000  000000    3$:     .WORD   0,0,0,0         ;FRACTIONAL RES.
       102124  000000
19903  102126  056340  000000  000000    4$:     .WORD   56340,0,0,1     ;INTEGER RES.
       102134  000001
19904  102136  000213                    7$:     213                     ;FPS BEFORE EXECUTION.
19905  102140  000204                            204                     ;FPS AFTER EXECUTION.
19906
19907                                    ;MODD TEST WITH EXPONENT OF THE RESULT = 56
```

```
19908  102142  004737  102576          MMM7:   JSR     PC,@#MODDSUB
19909  102146  056000  000000  000000  1$:     .WORD   56000,0,0,1     ;AC
       102154  000001
19910  102156  040340  000000  000000  2$:     .WORD   40340,0,0,0     ;FSRC
       102164  000000
19911  102166  040100  000000  000000  3$:     .WORD   40100,0,0,0     ;FRACTIONAL RES.
       102174  000000
19912  102176  056140  000000  000000  4$:     .WORD   56140,0,0,1     ;INTEGER RES.
       102204  000001
19913  102206  000213                  7$:     213                     ;FPS BEFORE EXECUTION.
19914  102210  000200                          200                     ;FPS AFTER EXECUTION.
19915
19916                                  ;MODD TEST WITH EXPONENT OF THE RESULT = 36
19917  102212  004737  102576          MMM8:   JSR     PC,@#MODDSUB
19918  102216  051177  177777  177777  1$:     .WORD   51177,-1,-1,-1  ;AC
       102224  177777
19919  102226  040200  000000  000000  2$:     .WORD   40200,0,0,0     ;FSRC
       102234  000000
19920  102236  040177  177760  000000  3$:     .WORD   40177,-20,0,0   ;FRACTIONAL RES.
       102244  000000
19921  102246  051177  177777  177760  4$:     .WORD   51177,-1,-20,0  ;INTEGER RES.
       102254  000000
19922  102256  000217                  7$:     217                     ;FPS BEFORE EXECUTION.
19923  102260  000200                          200                     ;FPS AFTER EXECUTION.
19924
19925                                  ;MODD TEST WITH EXPONENT OF THE RESULT = 30
19926  102262  004737  102576          MMM9:   JSR     PC,@#MODDSUB
19927  102266  040200  000000  000000  1$:     .WORD   40200,0,0,0     ;AC
       102274  000000
19928  102276  047577  177777          2$:     .WORD   47577,-1        ;FSRC
19929  102302  176000  000001                  .WORD   176000,1
19930  102306  031600  000000  000000  3$:     .WORD   31600,0,0,0     ;FRACTIONAL RES.
       102314  000000
19931  102316  047577  177777          4$:     .WORD   47577,-1        ;INTEGER RES.
19932  102322  176000  000000                  .WORD   176000,0
19933  102326  000200                  7$:     200                     ;FPS BEFORE EXECUTION.
19934  102330  000200                          200                     ;FPS AFTER EXECUTION.
19935
19936                                  ;MODD TEST WITH EXPONENT OF THE RESULT = 31
19937  102332  004737  102576          MMM10:  JSR     PC,@#MODDSUB
19938  102336  047777  177777          1$:     .WORD   47777,-1        ;AC
19939  102342  177000  000000                  .WORD   177000,0
19940  102346  040200  000000  000000  2$:     .WORD   40200,0,0,0     ;FSRC
       102354  000000
19941  102356  000000  000000  000000  3$:     .WORD   0,0,0,0         ;FRACTIONAL RES.
       102364  000000
19942  102366  047777  177777          4$:     .WORD   47777,-1        ;INTEGER RES.
19943  102372  177000  000000                  .WORD   177000,0
19944  102376  000213                  7$:     213                     ;FPS BEFORE EXECUTION.
19945  102400  000204                          204                     ;FPS AFTER EXECUTION.
19946
19947                                  ;MODD TEST WITH EXPONENT OF THE RESULT = 0
19948  102402  004737  102576          MMM11:  JSR     PC,@#MODDSUB
19949  102406  040200  000000  000000  1$:     .WORD   40200,0,0,0     ;AC
       102414  000000
19950  102416  040177  072727          2$:     .WORD   40177,72727     ;FSRC
```

```
19951  102422  127272  072727                        .WORD   127272,72727
19952  102426  040177  072727          3$:           .WORD   40177,72727     ;FRACTIONAL RES.
19953  102432  127272  072727                        .WORD   127272,72727
19954  102436  000000  000000  000000  4$:           .WORD   0,0,0,0         ;INTEGER RES.
       102444  000000
19955  102446  000200                  7$:           200                     ;FPS BEFORE EXECUTION.
19956  102450  000200                                200                     ;FPS AFTER EXECUTION.
19957
19958                                  ;MODD TEST WITH EXPONENT OF THE RESULT = -115
19959  102452  004737  102576          MMM12:  JSR     PC,@#MODDSUB
19960  102456  003377  177777          1$:           .WORD   3377,-1         ;AC
19961  102462  177777  052525                        .WORD   -1,52525
19962  102466  040200  000000  000000  2$:           .WORD   40200,0,0,0     ;FSRC
       102474  000000
19963  102476  003377  177777          3$:           .WORD   3377,-1         ;FRACTIONAL RES.
19964  102502  177777  052525                        .WORD   -1,52525
19965  102506  000000  000000  000000  4$:           .WORD   0,0,0,0         ;INTEGER RES.
       102514  000000
19966  102516  000200                  7$:           200                     ;FPS BEFORE EXECUTION.
19967  102520  000200                                200                     ;FPS AFTER EXECUTION.
19968
19969                                  ;MODD TEST WITH EXPONENT OF THE RESULT = -63, IN ROUND MODE.
19970  102522  004737  102576          MMM13:  JSR     PC,@#MODDSUB
19971  102526  040300  000000  000000  1$:           .WORD   40300,0,0,0     ;AC
       102534  000000
19972  102536  020200  000000  000000  2$:           .WORD   20200,0,0,1     ;FSRC
       102544  000001
19973  102546  020300  000000  000000  3$:           .WORD   20300,0,0,2     ;FRACTIONAL RES.
       102554  000002
19974  102556  000000  000000  000000  4$:           .WORD   0,0,0,0         ;INTEGER RES.
       102564  000000
19975  102566  000200                  7$:           200                     ;FPS BEFORE EXECUTION.
19976  102570  000200                                200                     ;FPS AFTER EXECUTION.
19977  102572  000137  102772          9$:           JMP     @#MMMDONE       ;GO TO THE NEXT TEST.
19978
19979                                  ;THIS SUBROUTINE, MODDSUB, IS CALLED TO SETUP THE
19980                                  ;OPERANDS, EXECUTE THE MODD INSTRUCTION AND CHECK THE RESULTS.
19981                                  ;IT IS CALLED THUS:
19982                                  ;
19983                                  ;               ACARG:  .WORD   X,X,X,X         ;AC OPERAND
19984                                  ;               FSRCARG:.WORD   X,X,X,X         ;FSRC OPERAND
19985                                  ;               FRES:   .WORD   X,X,X,X         ;FRACTIONAL RESULT
19986                                  ;               INTRES: .WORD   X,X,X,X         ;INTEGER RESULT
19987                                  ;               ERFRES: .WORD   X,X,X,X         ;ERROR FRACTION RESULT
19988                                  ;               ERINTRES:.WORD  X,X,X,X         ;ERROR INTEGER RESULT
19989                                  ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
19990                                  ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
19991                                  ;               ERR1:   ERROR   X               ;FRACTION ERROR
19992                                  ;                       BR      CONT
19993                                  ;               ERR2:   ERROR   X               ;INTEGER ERROR
19994                                  ;               CONT:                           ;RETURN ADDRESS
19995                                  ;
19996                                  ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC ARGUMENT). THE MODD
19997                                  ;INSTRUCTION IS EXECUTED. THEN THE RESULTS ARE RETRIEVED.
19998                                  ;THE FRACTION PART OF THE RESULT IS COMPARED WITH FRES. IF THIS IS CORRECT
19999                                  ;THEN THE INTEGER PART IS COMPARED WITH INTRES. IF BOTH OF THESE ARE CORRECT
```

```
20000                                   ;THEN THE FPS IS COMPARED WITH FPSA. AFTER EXECUTION IF NO ERRORS OCCURRED
20001                                   ;THEN MODDSUB WILL RETURN TO CONT. IF THE FPS WAS INCORRECT
20002                                   ;IT IS REPORTED HERE. IF THE FRACTION IS INCORRECT IT IS COMPARED WITH
20003                                   ;THE ANTICIPATED BAD FRACTION, ERFRES. IF THIS DOESN'T MATCH
20004                                   ;THE TRUE RESULT THEN THE ERROR IS REPORTED HERE. IF THE ANTICIPATED
20005                                   ;FAILURE MATCHES THE TRUE RESULT THEN  MODDSUB PASSES CONTROL TO THE
20006                                   ;ERROR CALL AT ERR1. LIKEWISE IF THE INTEGER PART OF THE RESULT IS
20007                                   ;NOT CORRECT THEN IT IS COMPARED WITH THE ANTICIPATED INTEGER
20008                                   ;FAILURE. IF THIS DOEN'T MATCH THEN THE ERROR IS REPORTED HERE.
20009                                   ;IF A MATCH IS MADE HOWEVER, MODDSUB WILL RETURN CONTROL TO THE ERROR
20010                                   ;CALL AT ERR2.
20011
20012  102576  012601                   MODDSUB:  MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS
20013  102600  012700  000200                 MOV     #200,R0         ;SET FD MODE.
20014  102604  170100                         LDFPS   R0
20015  102606  010100                         MOV     R1,R0           ;SET UP AC0
20016  102610  172410                         LDD     (R0),AC0
20017  102612  012700  101546                 MOV     #MODP1,R0       ;PUT A BACKROUND PATTERN INTO AC1.
20018  102616  172510                         LDD     (R0),AC1
20019  102620  016100  000040                 MOV     40(R1),R0       ;SET UP THE FPS.
20020  102624  170100                         LDFPS   R0
20021  102626  010100                         MOV     R1,R0           ;COMPUTE THE ADDRESS OF THE FSRC.
20022  102630  062700  000010                 ADD     #10,R0
20023
20024  102634  171410                   1$:   MODD    (R0),AC0        ;EXECUTE THE TEST INSTRUCTION.
20025
20026  102636  170204                         STFPS   R4              ;GET THE FPS.
20027  102640  012700  000200                 MOV     #200,R0         ;SET FD MODE.
20028  102644  170100                         LDFPS   R0
20029  102646  012700  102752                 MOV     #MODDT0,R0      ;GET THE FRACTIONAL RESULT.
20030  102652  174010                         STD     AC0,(R0)
20031  102654  012700  102762                 MOV     #MODDT1,R0      ;GET THE INTEGER RESULT.
20032  102660  174110                         STD     AC1,(R0)
20033  102662  012702  102752                 MOV     #MODDT0,R2      ;CHECK THE FRACTIONAL RESULT.
20034  102666  010103                         MOV     R1,R3
20035  102670  062703  000020                 ADD     #20,R3
20036  102674  012705  000004                 MOV     #4,R5
20037  102700  022223                   2$:   CMP     (R2)+,(R3)+
20038  102702  001401                         BEQ     4$
  (2)  102704  104000                         EMT                     ;
20039  102706  077504                   4$:   SOB     R5,2$
20040  102710  012702  102762                 MOV     #MODDT1,R2      ;CHECK THE INTEGER RESULT.
20041  102714  010103                         MOV     R1,R3
20042  102716  062703  000030                 ADD     #30,R3
20043  102722  012705  000004                 MOV     #4,R5
20044  102726  022223                   3$:   CMP     (R2)+,(R3)+
20045  102730  001401                         BEQ     5$
  (2)  102732  104000                         EMT                     ;
20046  102734  077504                   5$:   SOB     R5,3$
20047  102736  026104  000042                 CMP     42(R1),R4       ;CHECK THE FPS.
20048  102742  001401                         BEQ     9$
  (2)  102744  104000                         EMT                     ;
20049  102746  000161  000044           9$:   JMP     44(R1)          ;RETURN.
20050
20051  102752  000000  000000  000000 MODDTO: .WORD   0,0,0,0
       102760  000000
```

```
20052
20053  102762  000000  000000  000000  MODDT1: .WORD   0,0,0,0
       102770  000000
20054
20055  102772                          MMMDONE:
  (1)  102772  004767  021554                  JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
20056
20057
20058
20066
20067                                   ;*************************************************************************************
  (2)                                   ;TEST 507       UNDER\OVER FLOW, USING MODF WITH TRAPS DISABLED, TEST
  (3)                                   ;*************************************************************************************
  (2)  102776                           TS507:
20068
20069                                   ;UNDERFLOW TEST, WITH EXPONENT OF THE RESULT = -129, FIU = 1, FID = 1
20070  102776  004767  000214           MMM1:   JSR     PC,MODFOV
20071  103002  020123  045676           1$:     .WORD   20123,45676     ;AC
20072  103006  020200  000000           2$:     .WORD   20200,0         ;FSRC
20073  103012  000123  045676           3$:     .WORD   123,45676       ;FRACTIONAL RES.
20074  103016  000000  000000           4$:     .WORD   0,0             ;INTEGER RES.
20075  103022  042000                   7$:     42000                   ;FPS BEFORE EXECUTION.
20076  103024  142004                           142004                  ;FPS AFTER EXECUTION.
20077  103026  000012                           12                      ;FEC
20078  103030  104000                           EMT                     ;
20079
20080                                   ;UNDERFLOW EXP OF RESULT = -193, FIU = 0, FID = 1
20081  103032  004737  103216           MMM2:   JSR     PC,@#MODFOV
20082  103036  010200  000000           1$:     .WORD   10200,0         ;AC
20083  103042  010000  000000           2$:     .WORD   10000,0         ;FSRC
20084  103046  000000  000000           3$:     .WORD   0,0             ;FRACTIONAL RES.
20085  103052  000000  000000           4$:     .WORD   0,0             ;INTEGER RES.
20086  103056  005013                   7$:     5013                    ;FPS BEFORE EXECUTION.
20087  103060  005004                           5004                    ;FPS AFTER EXECUTION.
20088  103062  000012                           12                      ;FEC
20089  103064  000240                           NOP
20090
20091                                   ;OVERFLOW TEST WITH EXPONENT OF THE RESULT = 128, FIV = 1, FID = 1
20092  103066  004737  103216           MMM3:   JSR     PC,@#MODFOV
20093  103072  060052  125252           1$:      WORD   60052,125252     ;AC
20094  103076  060200  000000           2$:     .WORD   60200,0         ;FSRC
20095  103102  000000  000000           3$:     .WORD   0,0             ;FRACTIONAL RES.
20096  103106  000052  125252           4$:     .WORD   52,125252       ;INTEGER RES.
20097  103112  041000                   7$:     41000                   ;FPS BEFORE EXECUTION.
20098  103114  141006                           141006                  ;FPS AFTER EXECUTION.
20099  103116  000010                           10                      ;FEC
20100  103120                           8$:
  (2)  103120  104000                           EMT                     ;
20101                                   ;OVERFLOW TEST WITH EXPONENT OF THE RESULT = 130, FIV = 0, FID = 1
20102  103122  004737  103216           MMM4:   JSR     PC,@#MODFOV
20103  103126  060345  067654           1$:     .WORD   60345,67654     ;AC
20104  103132  060200  000000           2$:     .WORD   60200,0         ;FSRC
```

```
20105  103136  000000  000000          3$:     .WORD   0,0             ;FRACTIONAL RES.
20106  103142  000000  000000          4$:     .WORD   0,0             ;INTEGER RES.
20107  103146  006011                  7$:     6011                    ;FPS BEFORE EXECUTION.
20108  103150  006006                          6006                    ;FPS AFTER EXECUTION.
20109  103152  000010                          10                      ;FEC
20110  103154  000240                  8$:     NOP
20111                                  ;OVERFLOW TEST WITH EXPONENT OF THE RESULT = 128, RESULT NEGATIVE
20112                                  ;AND FIV = 1, FID = 1
20113  103156  004737  103216          MMM5:   JSR     PC,@#MODFOV
20114  103162  160252  125252          1$:     .WORD   160252,125252   ;AC
20115  103166  060000  000000          2$:     .WORD   60000,0         ;FSRC
20116  103172  000000  000000          3$:     .WORD   0,0             ;FRACTIONAL RES.
20117  103176  100052  125252          4$:     .WORD   100052,125252   ;INTEGER RES.
20118  103202  041000                  7$:     41000                   ;FPS BEFORE EXECUTION.
20119  103204  141006                          141006                  ;FPS AFTER EXECUTION.
20120  103206  000010                          10                      ;FEC
20121  103210                          8$:
  (2)  103210  104000                          EMT                     ;
20122  103212  000137  103432          9$:     JMP     @#MMMDONE       ;GO TO THE NEXT TEST.
20123
20124                                  ;THIS SUBROUTINE, MODFOV, IS CALLED TO SETUP THE
20125                                  ;OPERANDS, EXECUTE THE MODF INSTRUCTION AND CHECK THE RESULTS.
20126                                  ;IT IS CALLED THUS:
20127                                  ;
20128                                  ;               ACARG:   .WORD   X,X            ;AC OPERAND
20129                                  ;               FSRCARG:.WORD   X,X             ;FSRC OPERAND
20130                                  ;               FRES:    .WORD   X,X            ;FRACTIONAL RESULT
20131                                  ;               INTRES:  .WORD   X,X            ;INTEGER RESULT
20132                                  ;               ERFRES:  .WORD   X,X            ;ERROR FRACTION RESULT
20133                                  ;               ERINTRES:.WORD   X,X            ;ERROR INTEGER RESULT
20134                                  ;               FPSB:    .WORD   X              ;FPS BEFORE EXECUTION
20135                                  ;               FPSA:    .WORD   X              ;FPS AFTER EXECUTION
20136                                  ;               FEC:     .WORD   X              ;FEC
20137                                  ;               ERR1:    ERROR   X              ;FEC ERROR
20138                                  ;                        BR      CONT
20139                                  ;               ERR2:    ERROR   X              ;INTEGER ERROR
20140                                  ;               CONT:                           ;RETURN ADDRESS
20141                                  ;
20142                                  ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC ARGUMENT). THE MODF
20143                                  ;INSTRUCTION IS EXECUTED. THEN THE RESULTS ARE RETRIEVED.
20144                                  ;THE FRACTION PART OF THE RESULT IS COMPARED WITH FRES. IF THIS IS CORRECT
20145                                  ;THEN THE INTEGER PART IS COMPARED WITH INTRES. IF BOTH OF THESE ARE CORRECT
20146                                  ;THEN THE FPS IS COMPARED WITH FPSA. AFTER EXECUTION IF NO ERRORS OCCURRED
20147                                  ;THEN MODFOV WILL RETURN TO CONT. IF THE FPS WAS INCORRECT
20148                                  ;IT IS REPORTED HERE. IF THE FRACTION IS INCORRECT IT IS COMPARED WITH
20149                                  ;THE ANTICIPATED BAD FRACTION, ERFRES. IF THIS DOESN'T MATCH
20150                                  ;THE TRUE RESULT THEN THE ERROR IS REPORTED HERE. IF THE ANTICIPATED
20151                                  ;FAILURE MATCHES THE TRUE RESULT THEN  MODFOV PASSES CONTROL TO THE
20152                                  ;ERROR CALL AT ERR1. LIKEWISE IF THE INTEGER PART OF THE RESULT IS
20153                                  ;NOT CORRECT THEN IT IS COMPARED WITH THE ANTICIPATED INTEGER
20154                                  ;FAILURE. IF THIS DOEN'T MATCH THEN THE ERROR IS REPORTED HERE.
20155                                  ;IF A MATCH IS MADE HOWEVER, MODFOV WILL RETURN CONTROL TO THE ERROR
20156                                  ;CALL AT ERR2.
20157
20158  103216  012601                  MODFOV: MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS
20159  103220  012700  000200                  MOV     #200,R0         ;SET FD MODE.
```

```
20160  103224  170100                          LDFPS   R0
20161  103226  010100                          MOV     R1,R0           ;SET UP AC0
20162  103230  172410                          LDD     (R0),AC0
20163  103232  012700  101546                  MOV     #MODP1,R0       ;PUT A BACKROUND PATTERN INTO AC1.
20164  103236  172510                          LDD     (R0),AC1
20165  103240  016100  000020                  MOV     20(R1),R0       ;SET UP THE FPS.
20166  103244  170100                          LDFPS   R0
20167  103246  010100                          MOV     R1,R0           ;COMPUTE THE ADDRESS OF THE FSRC.
20168  103250  062700  000004                  ADD     #4,R0
20169
20170  103254  171410                  1$:     MODF    (R0),AC0        ;EXECUTE THE TEST INSTRUCTION.
20171
20172  103256  170204                          STFPS   R4              ;GET THE FPS.
20173  103260  170305                          STST    R5              ;GET FEC.
20174  103262  012700  000200                  MOV     #200,R0         ;SET FD MODE.
20175  103266  170100                          LDFPS   R0
20176  103270  012700  103412                  MOV     #MODFD0,R0      ;GET THE FRACTIONAL RESULT.
20177  103274  174010                          STD     AC0,(R0)
20178  103276  012700  103422                  MOV     #MODFD1,R0      ;GET THE INTEGER RESULT.
20179  103302  174110                          STD     AC1,(R0)
20180  103304  012702  103412                  MOV     #MODFD0,R2      ;CHECK THE FRACTIONAL RESULT.
20181  103310  026112  000010                  CMP     10(R1),(R2)
20182  103314  001401                          BEQ     2$
  (2)  103316  104000                          EMT                     ;
20183  103320  026162  000012  000002  2$:     CMP     12(R1),2(R2)
20184  103326  001401                          BEQ     3$
  (2)  103330  104000                          EMT                     ;
20185  103332  012702  103422          3$:     MOV     #MODFD1,R2      ;CHECK THE INTEGER RESULT.
20186  103336  026112  000014                  CMP     14(R1),(R2)
20187  103342  001401                          BEQ     4$
  (2)  103344  104000                          EMT                     ;
20188  103346  026162  000016  000002  4$:     CMP     16(R1),2(R2)
20189  103354  001401                          BEQ     5$
  (2)  103356  104000                          EMT                     ;
20190  103360  026104  000022          5$:     CMP     22(R1),R4       ;CHECK THE FPS.
20191  103364  001401                          BEQ     6$
  (2)  103366  104000                          EMT                     ;
20192  103370  026105  000024          6$:     CMP     24(R1),R5       ;CHECK THE FEC.
20193  103374  001002                          BNE     25$             ;BRANCH IF INCORRECT.
20194
20195  103376  000161  000030          9$:     JMP     30(R1)          ;RETURN.
20196                                  ;REPORT FEC ERROR.
20197  103402  010102                  25$:    MOV     R1,R2
20198  103404  062702  000026                  ADD     #26,R2
20199  103410  000112                          JMP     (R2)
20200
20201  103412  000000  000000  000000  MODFD0: .WORD   0,0,0,0
       103420  000000
20202
20203  103422  000000  000000  000000  MODFD1: .WORD   0,0,0,0
       103430  000000
20204
20205  103432                          MMMDONE:
  (1)  103432  004767  021114                  JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
```

```
  (1)                                                   ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                   ;THE USER TYPED CONTROL G?).
20206
20207
20208
20216
20217                                   ;*****************************************************************************
  (2)                                   ;TEST 510       UNDER\OVER FLOW, USING MODD WITH TRAPS DISABLED, TEST
  (3)                                   ;*****************************************************************************
  (2) 103436                            TS510:
20218
20219                                   ;UNDERFLOW TEST WITH EXPONENT OF THE RESULT = -129, FIU = 1, FID = 1
20220 103436  004737  103722            NNN1:   JSR     PC,@#MODDOV
20221 103442  020252  125252            1$:     .WORD   20252,125252    ;AC
20222 103446  125252  125252                    .WORD   125252,125252
20223 103452  020100  000000  000000    2$:     .WORD   20100,0,0,0     ;FSRC
      103460  000000
20224 103462  000177  177777  177777    3$:     .WORD   177,-1,-1,-1    ;FRACTIONAL RES.
      103470  177777
20225 103472  000000  000000  000000    4$:     .WORD   0,0,0,0         ;INTEGER RES.
      103500  000000
20226 103502  042200                    7$:     42200                   ;FPS BEFORE EXECUTION.
20227 103504  142204                            142204                  ;FPS AFTER EXECUTION.
20228 103506  000012                            12                      ;FEC
20229 103510                            8$:
  (2) 103510  104000                            EMT                     ;
20230                                   ;UNDERFLOW TEST WITH EXPONENT OF THE RESULT = -193, FIU = 0, FID = 1
20231 103512  004737  103722            NNN2:   JSR     PC,@#MODDOV
20232 103516  010000  000000            1$:     .WORD   10000,0         ;AC
20233 103522  123456  000000                    .WORD   123456,0
20234 103526  010200  000000  000000    2$:     .WORD   10200,0,0,0     ;FSRC
      103534  000000
20235 103536  000000  000000  000000    3$:     .WORD   0,0,0,0         ;FRACTIONAL RES.
      103544  000000
20236 103546  000000  000000  000000    4$:     .WORD   0,0,0,0         ;INTEGER RES.
      103554  000000
20237 103556  005213                    7$:     5213                    ;FPS BEFORE EXECUTION.
20238 103560  005204                            5204                    ;FPS AFTER EXECUTION.
20239 103562  000012                            12
20240 103564  000240                    8$:     NOP
20241                                   ;OVERFLOW TEST WITH EXPONENT OF THE RESULT = 128, FIV = 1, FID = 1
20242 103566  004737  103722            NNN3:   JSR     PC,@#MODDOV
20243 103572  060252  125252            1$:     .WORD   60252,125252    ;AC
20244 103576  125252  125252                    .WORD   125252,125252
20245 103602  060100  000000  000000    2$:     .WORD   60100,0,0,0     ;FSRC
      103610  000000
20246 103612  000000  000000  000000    3$:     .WORD   0,0,0,0         ;FRACTIONAL RES.
      103620  000000
20247 103622  000177  177777  177777    4$:     .WORD   177,-1,-1,-1    ;INTEGER RES.
      103630  177777
20248 103632  041200                    7$:     41200                   ;FPS BEFORE EXECUTION.
20249 103634  141206                            141206                  ;FPS AFTER EXECUTION.
20250 103636  000010                            10                      ;FEC
20251 103640                            8$:
  (2) 103640  104000                            EMT                     ;
20252                                   ;OVERFLOW TEST WITH EXPONENT OF THE RESULT = 130, FIV = 0, FID = 1
```

```
20253  103642  004737  103722          NNN4:   JSR     PC,@#MODDOV
20254  103646  060200  000000          1$:     .WORD   60200,0         ;AC
20255  103652  125252  000000                  .WORD   125252,0
20256  103656  060200  000000  000000  2$:     .WORD   60200,0,0,0     ;FSRC
       103664  000000
20257  103666  000000  000000  000000  3$:     .WORD   0,0,0,0         ;FRACTIONAL RES.
       103674  000000
20258  103676  000000  000000  000000  4$:     .WORD   0,0,0,0         ;INTEGER RES.
       103704  000000
20259  103706  006211                  7$:     6211                    ;FPS BEFORE EXECUTION.
20260  103710  006206                          6206                    ;FPS AFTER EXECUTION.
20261  103712  000010                          10                      ;FEC
20262  103714  000240                  8$:     NOP
20263  103716  000137  104136          9$:     JMP     @#NNNDONE       ;GO TO NEXT TEST.
20264
20265                                  ;THIS SUBROUTINE, MODDOV, IS CALLED TO SETUP THE
20266                                  ;OPERANDS, EXECUTE THE MODD INSTRUCTION AND CHECK THE RESULTS.
20267                                  ;IT IS CALLED THUS:
20268                                  ;
20269                                  ;               ACARG:  .WORD   X,X,X,X         ;AC OPERAND
20270                                  ;               FSRCARG:.WORD   X,X,X,X         ;FSRC OPERAND
20271                                  ;               FRES:   .WORD   X,X,X,X         ;FRACTIONAL RESULT
20272                                  ;               INTRES: .WORD   X,X,X,X         ;INTEGER RESULT
20273                                  ;               ERFRES: .WORD   X,X,X,X         ;ERROR FRACTION RESULT
20274                                  ;               ERINTRES:.WORD  X,X,X,X         ;ERROR INTEGER RESULT
20275                                  ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
20276                                  ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
20277                                  ;               ERR1:   ERROR   X               ;FRACTION ERROR
20278                                  ;                       BR      CONT
20279                                  ;               ERR2:   ERROR   X               ;INTEGER ERROR
20280                                  ;               CONT:                           ;RETURN ADDRESS
20281                                  ;
20282                                  ;THE OPERANDS ARE SET UP (USING AC0 FOR THE AC ARGUMENT). THE MODD
20283                                  ;INSTRUCTION IS EXECUTED. THEN THE RESULTS ARE RETRIEVED.
20284                                  ;THE FRACTION PART OF THE RESULT IS COMPARED WITH FRES. IF THIS IS CORRECT
20285                                  ;THEN THE INTEGER PART IS COMPARED WITH INTRES. IF BOTH OF THESE ARE CORRECT
20286                                  ;THEN THE FPS IS COMPARED WITH FPSA. AFTER EXECUTION IF NO ERRORS OCCURRED
20287                                  ;THEN MODDOV WILL RETURN TO CONT. IF THE FPS WAS INCORRECT
20288                                  ;IT IS REPORTED HERE. IF THE FRACTION IS INCORRECT IT IS COMPARED WITH
20289                                  ;THE ANTICIPATED BAD FRACTION, ERFRES. IF THIS DOESN'T MATCH
20290                                  ;THE TRUE RESULT THEN THE ERROR IS REPORTED HERE. IF THE ANTICIPATED
20291                                  ;FAILURE MATCHES THE TRUE RESULT THEN  MODDOV PASSES CONTROL TO THE
20292                                  ;ERROR CALL AT ERR1. LIKEWISE IF THE INTEGER PART OF THE RESULT IS
20293                                  ;NOT CORRECT THEN IT IS COMPARED WITH THE ANTICIPATED INTEGER
20294                                  ;FAILURE. IF THIS DOEN'T MATCH THEN THE ERROR IS REPORTED HERE.
20295                                  ;IF A MATCH IS MADE HOWEVER, MODDOV WILL RETURN CONTROL TO THE ERROR
20296                                  ;CALL AT ERR2.
20297
20298  103722  012601                  MODDOV: MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS
20299  103724  012700  000200                  MOV     #200,R0         ;SET FD MODE.
20300  103730  170100                          LDFPS   R0
20301  103732  010100                          MOV     R1,R0           ;SET UP AC0
20302  103734  172410                          LDD     (R0),AC0
20303  103736  012700  101546                  MOV     #MODP1,R0       ;PUT A BACKROUND PATTERN INTO AC1.
20304  103742  172510                          LDD     (R0),AC1
20305  103744  016100  000040                  MOV     40(R1),R0       ;SET UP THE FPS.
```

```
20306  103750  170100                          LDFPS   R0
20307  103752  010100                          MOV     R1,R0           ;COMPUTE THE ADDRESS OF THE FSRC.
20308  103754  062700  000010                  ADD     #10,R0
20309
20310  103760  171410                  1$:     MODD    (R0),AC0        ;EXECUTE THE TEST INSTRUCTION.
20311
20312  103762  170305                          STST    R5              ;GET THE FPS.
20313  103764  170204                          STFPS   R4              ;GET THE FPS.
20314  103766  012700  000200                  MOV     #200,R0         ;SET FD MODE.
20315  103772  170100                          LDFPS   R0
20316  103774  012700  104116                  MOV     #MODDD0,R0      ;GET THE FRACTIONAL RESULT.
20317  104000  174010                          STD     AC0,(R0)
20318  104002  012700  104126                  MOV     #MODDD1,R0      ;GET THE INTEGER RESULT.
20319  104006  174110                          STD     AC1,(R0)
20320  104010  012702  104116                  MOV     #MODDD0,R2      ;CHECK THE FRACTIONAL RESULT.
20321  104014  010103                          MOV     R1,R3
20322  104016  062703  000020                  ADD     #20,R3
20323  104022  012700  000004                  MOV     #4,R0
20324  104026  022223                  2$:     CMP     (R2)+,(R3)+
20325  104030  001401                          BEQ     4$
  (2)  104032  104000                          EMT                     ;
20326  104034  077004                  4$:     SOB     R0,2$
20327  104036  012702  104126                  MOV     #MODDD1,R2      ;CHECK THE INTEGER RESULT
20328  104042  010103                          MOV     R1,R3
20329  104044  062703  000030                  ADD     #30,R3
20330  104050  012700  000004                  MOV     #4,R0
20331  104054  022223                  3$:     CMP     (R2)+,(R3)+
20332  104056  001401                          BEQ     5$
  (2)  104060  104000                          EMT                     ;
20333  104062  077004                  5$:     SOB     R0,3$
20334  104064  026104  000042                  CMP     42(R1),R4       ;CHECK THE FPS.
20335  104070  001401                          BEQ     6$
  (2)  104072  104000                          EMT                     ;
20336  104074  026105  000044          6$:     CMP     44(R1),R5       ;CHECK THE FEC.
20337  104100  001002                          BNE     25$
20338
20339  104102  000161  000050          9$:     JMP     50(R1)          ;RETURN.
20340                                  ;REPORT FEC ERROR.
20341  104106  010102                  25$:    MOV     R1,R2
20342  104110  062702  000046                  ADD     #46,R2
20343  104114  000112                          JMP     (R2)
20344
20345  104116  000000  000000  000000  MODDD0: .WORD   0,0,0,0
       104124  000000
20346
20347  104126  000000  000000  000000  MODDD1: .WORD   0,0,0,0
       104134  C00000
20348
20349  104136                          NNNDONE:
  (1)  104136  004767  020410                  JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
20350
20356
```

```
20357                                   ;*****************************************************************************
  (2)                                   ;TEST 511       MORE MICROCODES COVERAGE
  (3)                                   ;*****************************************************************************
  (2) 104142                            TS511:
20358 104142  012737  104156  000244    XT1:    MOV     #XT1A,@#244
20359 104150  170227  000000            STFPS   #0
20360 104154  000401                    BR      XT2
20361 104156                            XT1A:
  (2) 104156  104000                    EMT                             ;
20362
20363 104160  012700  177777            XT2:    MOV     #-1,R0
20364 104164  170127  000000            LDFPS   #0
20365 104170  170200                    STFPS   R0
20366 104172  005700                    TST     R0
20367 104174  001401                    BEQ     XT2A
  (2) 104176  104000                    EMT                             ;
20368 104200  012700  104734            XT2A:   MOV     #XPAT0,R0
20369 104204  172440                    LDF     -(R0),AC0
20370 104206  022700  104730            CMP     #XPAT0-4,R0
20371 104212  001401                    BEQ     XT2B
  (2) 104214  104000                    EMT                             ;
20372 104216  170200                    XT2B:   STFPS   R0
20373 104220  022700  000004            CMP     #4,R0                   ;CHECK IF FZ IS SET?
20374 104224  001401                    BEQ     XT3
  (2) 104226  104000                    EMT                             ;
20375 104230  170127  000000            XT3:    LDFPS   #0
20376 104234  012700  104734            MOV     #XPAT0,R0
20377 104240  174040                    STF     AC0,-(R0)
20378 104242  022700  104730            CMP     #XPAT0-4,R0
20379 104246  001401                    BEQ     XT3A
  (2) 104250  104000                    EMT                             ;
20380 104252  170200                    XT3A:   STFPS   R0
20381 104254  005700                    TST     R0
20382 104256  001401                    BEQ     XT4
  (2) 104260  104000                    EMT                             ;
20383
20384 104262  170127  000000    XT4:    LDFPS   #0
20385 104266  012737  104312  000244    MOV     #XT4A,@#244
20386 104274  170127  004000            LDFPS   #04000                  ;INTRPT ON UNDEFINED VARIABLE
20387 104300  172437  104734            LDF     @#XPAT0,AC0
20388 104304  174437  104764            DIVF    @#XPAT3,AC0             ;GET UNDEFINED VARIABLE, _0
20389 104310  104000                    EMT                             ;
20390 104312  170200            XT4A:   STFPS   R0
20391 104314  022700  104004            CMP     #104004,R0              ;CHECK: FER,FIUV,FZ ARE SET?
20392 104320  001401                    BEQ     XT4B
  (2) 104322  104000                    EMT                             ;
20393 104324  012700  104724    XT4B:   MOV     #XBUF,R0
20394 104330  174010                    STF     AC0,(R0)
20395 104332  005737  104724            TST     @#XBUF
20396 104336  001401                    BEQ     XT5
  (2) 104340  104000                    EMT                             ;
20397
20398 104342  012737  104362  000244    XT5:    MOV     #XT5A,@#244
20399 104350  170127  004000            LDFPS   #04000                  ;INTRPT ON UNDEFINED VARIBALE
20400 104354  177437  104764            LDCDF   @#XPAT3,AC0             ;GET UNDEFINED VARIABLE, _0
20401 104360  104000                    EMT                             ;
```

```
20402  104362  170200                 XT5A:   STFPS   R0
20403  104364  022700  104014                 CMP     #104014,R0      ;CHECK: FER,FIUV,FN,FZ ARE SET?
20404  104370  001401                         BEQ     XT5B
  (2)  104372  104000                         EMT                     ;
20405  104374  012700  104724         XT5B:   MOV     #XBUF,R0
20406  104400  174010                         STF     AC0,(R0)
20407  104402  005737  104724                 TST     @#XBUF
20408  104406  001401                         BEQ     XT6
  (2)  104410  104000                         EMT                     ;
20409
20410  104412  012737  104436  000244 XT6:    MOV     #XT6A,@#244
20411  104420  170127  004000                 LDFPS   #04000          ;INTRPT ON UNDEFINED VARIBALE
20412  104424  172437  104734                 LDF     @#XPAT0,AC0
20413  104430  172037  104764                 ADDF    @#XPAT3,AC0
20414  104434  104000                         EMT                     ;
20415  104436  170200                 XT6A:   STFPS   R0
20416  104440  022700  104004                 CMP     #104004,R0      ;CHECK: FER,FIUV,FZ ARE SET?
20417  104444  001401                         BEQ     XT6B
  (2)  104446  104000                         EMT                     ;
20418  104450  012700  104724         XT6B:   MOV     #XBUF,R0
20419  104454  174010                         STF     AC0,(R0)
20420  104456  005737  104724                 TST     @#XBUF
20421  104462  001401                         BEQ     XT7
  (2)  104464  104000                         EMT                     ;
20422
20423  104466  170127  000000         XT7:    LDFPS   #0
20424  104472  172437  104774                 LDF     @#XPAT4,AC0
20425  104476  175437  105024                 STCFI   AC0,@#XPAT0
20426  104502  022737  000002  105024         CMP     #2,@#XPAT0      ;CHECK DATA
20427  104510  001401                         BEQ     XT8
  (2)  104512  104000                         EMT                     ;
20428
20429  104514  170127  000100         XT8:    LDFPS   #100            ;SET FL
20430  104520  172437  104774                 LDF     @#XPAT4,AC0
20431  104524  175467  000274                 STCFI   AC0,XPAT0
20432  104530  022737  000002  105026         CMP     #2,@#XPAT0+2
20433  104536  001401                         BEQ     XT9
  (2)  104540  104000                         EMT                     ;
20434
20435                         ;START OF FPP2
20436  104542  170127  000000         XT9:    LDFPS   #0
20437  104546  172437  104734                 LDF     @#XPAT0,AC0
20438  104552  172037  104774                 ADDF    @#XPAT4,AC0
20439  104556  170200                         STFPS   R0
20440  104560  005700                         TST     R0
20441  104562  001401                         BEQ     XT10
  (2)  104564  104000                         EMT                     ;
20442
20443  104566  170127  000000         XT10:   LDFPS   #0
20444  104572  172437  104774                 LDF     @#XPAT4,AC0
20445  104576  173037  104774                 SUBF    @#XPAT4,AC0
20446  104602  170200                         STFPS   R0
20447  104604  022700  000004                 CMP     #4,R0
20448  104610  001401                         BEQ     XT11
  (2)  104612  104000                         EMT                     ;
20449
```

```
20450  104614  170127  000000          XT11:   LDFPS   #0
20451  104620  172437  104774                  LDF     @#XPAT4,AC0
20452  104624  173437  104774                  CMPF    @#XPAT4,AC0
20453  104630  170200                          STFPS   R0
20454  104632  022700  000004                  CMP     #4,R0           ;CHECK IF FZ IS SET?
20455  104636  001401                          BEQ     XT12
  (2)  104640  104000                          EMT                     ;
20456
20457  104642  170127  000000          XT12:   LDFPS   #0
20458  104646  172437  104774                  LDF     @#XPAT4,AC0
20459  104652  174437  104754                  DIVF    @#XPAT2,AC0
20460  104656  012700  104724                  MOV     #XBUF,R0
20461  104662  174010                          STF     AC0,(R0)
20462  104664  022737  040176  104724          CMP     #040176,@#XBUF  ;CHECK DATA
20463  104672  001401                          BEQ     XT13
  (2)  104674  104000                          EMT                     ;
20464
20465  104676  170127  000000          XT13:   LDFPS   #0
20466  104702  172437  105004                  LDF     @#XPAT5,AC0
20467  104706  174437  105014                  DIVF    @#XPAT6,AC0
20468  104712  170200                          STFPS   R0
20469  104714  022700  000004                  CMP     #4,R0
20470  104720  001445                          BEQ     XTDONE
  (2)  104722  104000                          EMT                     ;
20471
20472
20473  104724  000000  000000  000000  XBUF:   .WORD   0,0,0,0
       104732  000000
20474  104734  000000  000000  000000  XPAT0:  .WORD   0,0,0,0
       104742  000000
20475  104744  000001  000001  000001  XPAT1:  .WORD   1,1,1,1
       104752  000001
20476  104754  040401  000000  000000  XPAT2:  .WORD   40401,0,0,0
       104762  000000
20477  104764  100000  000000  000000  XPAT3:  .WORD   100000,0,0,0
       104772  000000
20478  104774  040400  000000  000000  XPAT4:  .WORD   040400,0,0,0
       105002  000000
20479  105004  000207  000000  000000  XPAT5:  .WORD   207,0,0,0
       105012  000000
20480  105014  077007  000000  000000  XPAT6:  .WORD   77007,0,0,0
       105022  000000
20481  105024  000000  000000  000000  XPAT0:  .WORD   0,0,0,0
       105032  000000
20482
20483  105034                          XTDONE:
  (1)  105034  004767  017512                  JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
20484
20485
20486
20492
20493                           ;*****************************************************************************
```

```
  (2)                                   ;TEST 512       STF WITH ILLEGAL ACCUMULATOR TEST
  (3)                                   ;*****************************************************************************
  (2)  105040                           TS512:
20494
20495  105040  005000                          CLR     R0              ;SET THE FPS.
20496  105042  170100                          LDFPS   R0
20497
20498  105044  012737  105064  000244          MOV     #OOOT,@#FPVECT  ;SET UP FOR FP TRAPS.
20499  105052  012737  105060  037576          MOV     #1$,@#$TMP2
20500
20501  105060  174007                 1$:      STF     AC0,AC7         ;THIS TEST INSTRUCTION SHOULD
20502                                                                  ;CAUSE A TRAP.
20503
20504                                 ;REPORT FAILURE OF USE OF ILLEGAL ACCUMULATOR 7 TO CAUSE AN FPP TRAP.
20505  105062                         0002:
20506  105062  104000                          EMT                     ;INSTRUCTION DID NOT TRAP
20507
20508                                 ;TRAP TO OOOT, HERE, WHEN THE EXPECTED ERROR OCCURS.
20509  105064  011600                 OOOT:    MOV     (SP),R0         ;MAKE SURE THE ERROR OCCURRED
20510  105066  022700  105062                 CMP     #0002,R0        ;AT THE CORRECT ADDRESS.
20511  105072  001420                          BEQ     TS513
  (3)  105074  104000                          EMT                     ;FLOATING POINT TRAP DID NOT OPERATE RIGHT
20512
20513  105076  170204                 0003:    STFPS   R4              ;GET FPS.
20514  105100  170305                          STST    R5              ;GET FEC.
20515  105102  012702  100000                 MOV     #100000,R2      ;EXPECTED FPS
20516  105106  012703  000002                 MOV     #2,R3           ;EXPECTED FEC
20517  105112  022626                          CMP     (SP)+,(SP)+     ;RESET THE STACK.
20518
20519  105114  020204                          CMP     R2,R4           ;WAS FPS CORRECT?
20520  105116  001401                          BEQ     0004
  (2)  105120  104000                          EMT                     ;FPS INCORRECTLY SET AFTER USE OF ILLEGAL ACC
20521  105122  020305                 0004:    CMP     R3,R5           ;WAS THE FEC CORRECT?
20522  105124  001401                          BEQ     OOODONE
  (2)  105126  104000                          EMT                     ;INCORRECT FEC AFTER USE OF ILLEGAL ACC
20523
20524  105130                         OOODONE:
  (1)  105130  004767  017416                  JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
20525
20526
20527
20533
20534                                 ;*****************************************************************************
  (2)                                 ;TEST 513       FDST MODE 1, FLOATING MODE, TEST
  (3)                                 ;*****************************************************************************
  (2)  105134                           TS513:
20535
20536
20537  105134  012700  177777                  MOV     #-1,R0          ;SET UP A BACKROUND PATTERN IN THE
20538  105140  012701  105250                  MOV     #PPPBF0,R1      ;INPUT BUFFER.
20539  105144  012702  000014                  MOV     #14,R2
20540  105150  010021                 PPP2:    MOV     R0,(R1)+
```

```
20541   105152  077202                  SOB     R2,PPP2
20542
20543   105154  012700  000200          MOV     #200,R0         ;SET FD MODE.
20544   105160  170100                  LDFPS   R0
20545   105162  012700  105300          MOV     #PPPTP1,R0      ;PUT TEST DATA INTO AC0.
20546   105166  172410                  LDD     (R0),AC0
20547
```

```
20549  105170  012700  105264          MOV     #PPPBF1,R0      ;FDST ADDRESS.
20550  105174  005002                  CLR     R2              ;CLEAR THE FPS.
20551  105176  170102                  LDFPS   R2
20552
20553  105200  174010          PPP3:   STF     AC0,(R0)        ;TEST INSTRUCTION.
20554
20555  105202  022700  105264          CMP     #PPPBF1,R0      ;WAS R0 MODIFIED DURING EXECUTION?
20556  105206  001401                  BEQ     PPP4
  (2)  105210  104000                  EMT                     ;R0 MODIFIED
20557
20558  105212  012700  105264  PPP4:   MOV     #PPPBF1,R0      ;CHECK THE DATA IN THE OUTPUT BUFFER.
20559  105216  012701  105300          MOV     #PPPTP1,R1
20560  105222  022021                  CMP     (R0)+,(R1)+
20561  105224  001031                  BNE     PPP10           ;BRANCH IF INCORRECT.
20562  105226  022011                  CMP     (R0)+,(R1)
20563  105230  001027                  BNE     PPP10           ;BRANCH IF INCORRECT.
20564  105232  022720  177777          CMP     #-1,(R0)+       ;WAS FLOATING MODE USED?
20565  105236  001024                  BNE     PPP10           ;BRANCH IF NOT.
20566  105240  022710  177777          CMP     #-1,(R0)
20567  105244  001021                  BNE     PPP10
20568  105246  000421                  BR      PPPDONE ;GO TO NEXT TEST.
20569
20570  105250  177777  177777  177777  PPPBF0: .WORD   -1,-1,-1,-1,-1,-1
       105256  177777  177777  177777
20571
20572  105264  177777  177777  177777  PPPBF1: .WORD   -1,-1,-1,-1,-1,-1
       105272  177777  177777  177777
20573
20574  105300  123456  023456          PPPTP1: .WORD   123456,23456
20575  105304  034567  045671                  .WORD   34567,45671
20576
20577  105310                          PPP10:
  (2)  105310  104000                          EMT                     ;
20578  105312                          PPPDONE:
  (1)  105312  004767  017234                  JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
20579
20580
20581
20587
20588                                   ;*************************************************************************************
  (2)                                   ;TEST 514       FDST MODE 2 TEST
  (3)                                   ;*************************************************************************************
  (2)  105316                           TS514:
20589
20590                                   ;FIRST TEST STF.
20591
20592  105316  012700  177777                  MOV     #-1,R0  ;SET UP THE OUTPUT BUFFER.
20593  105322  012701  105432                  MOV     #QQQBF0,R1
20594  105326  012702  000014                  MOV     #14,R2
20595  105332  010021                  QQQ2:   MOV     R0,(R1)+
20596  105334  077202                          SOB     R2,QQQ2
20597
```

```
20598  105336  012700  000200              MOV     #200,R0         ;SET FD MODE.
20599  105342  170100                      LDFPS   R0
20600  105344  012700  105462              MOV     #QQQTP1,R0      ;SETUP AC0.
20601  105350  172410                      LDD     (R0),AC0
20602
20603  105352  012700  105446              MOV     #QQQBF1,R0      ;FDST ADDRESS.
20604  105356  005002                      CLR     R2
20605  105360  170102                      LDFPS   R2              ;SET FPS.
20606  105362  174020              QQQ3:   STF     AC0,(R0)+       ;TEST INSTRUCTION.
20607
20608  105364  022700  105452              CMP     #QQQBF1+4,R0    ;WAS R0 INCREMENTED BY 4 PROPERLY?
20609
20610  105370  001401                      BEQ     QQQ4
  (2)  105372  104000                      EMT                     ;REPORT R0 INCORRECT AFTER FDST MODE 2
20611  105374  012700  105446      QQQ4:   MOV     #QQQBF1,R0      ;WAS THE OUTPUT DATA CORRECT?
20612  105400  012701  105462              MOV     #QQQTP1,R1
20613  105404  022021                      CMP     (R0)+,(R1)+
20614  105406  001031                      BNE     QQQ10           ;BRANCH IF INCORRECT.
20615  105410  022021                      CMP     (R0)+,(R1)+
20616  105412  001027                      BNE     QQQ10           ;BRANCH IF INCORRECT.
20617  105414  022027  177777              CMP     (R0)+,#-1       ;SEE IF ANY OTHER DATA BUFFER WORDS WERE MODIFIED.
20618  105420  001024                      BNE     QQQ10           ;BRANCH IF INCORRECT.
20619  105422  022027  177777              CMP     (R0)+,#-1
20620  105426  001021                      BNE     QQQ10           ;BRANCH IF INCORRECT.
20621  105430  000421                      BR      QQQ20
20622  105432  177777  177777  177777  QQQBF0: .WORD   -1,-1,-1,-1,-1,-1
       105440  177777  177777  177777
20623  105446  177777  177777  177777  QQQBF1: .WORD   -1,-1,-1,-1,-1,-1
       105454  177777  177777  177777
20624  105462  076543              QQQTP1: 76543
20625  105464  065432                      65432
20626  105466  054321                      54321
20627  105470  043210                      43210
20628                              ;REPORT OUTPUT DATA INCORRECT:
20629  105472                      QQQ10:
  (2)  105472  104000                      EMT                     ;
20630
20631                              ;NOW TEST STD MODE 2.
20632
20633  105474  012700  105432      QQQ20:  MOV     #QQQBF0,R0      ;SET UP DEFAULT INPUT DATA BUFFER.
20634  105500  010001                      MOV     R0,R1
20635  105502  012702  000014              MOV     #14,R2
20636  105506  010021              QQQ22:  MOV     R0,(R1)+
20637  105510  077202                      SOB     R2,QQQ22
20638  105512  012700  000200              MOV     #200,R0         ;ENTER FLOATING DOUBLE MODE.
20639  105516  170100                      LDFPS   R0
20640  105520  012700  105462              MOV     #QQQTP1,R0      ;LOAD AC0.
20641  105524  172410                      LDD     (R0),AC0
20642  105526  012700  105446              MOV     #QQQBF1,R0      ;SET DESTINATION ADDRESS.
20643  105532  012737  105540  037576      MOV     #QQQ23,@#$TMP2
20644  105540  174020              QQQ23:  STD     AC0,(R0)+       ;TEST INSTRUCTION.
20645  105542  022700  105456              CMP     #QQQBF1+10,R0   ;WAS R0 INCREMENTED BY 10 CORRECTLY?
20646  105546  001401                      BEQ     QQQ24
  (2)  105550  104000                      EMT                     ;REPORT R0 INCORRECTLY INCREMENTED
20647  105552  012700  105446      QQQ24:  MOV     #QQQBF1,R0      ;DID THE DATA REACH THE OUTPUT BUFFER CORRECTLY?
20648  105556  012701  105462              MOV     #QQQTP1,R1
```

```
20649  105552  012702  000004            MOV     #4,R2
20650  105556  022021                1$: CMP     (R0)+,(R1)+
20651  105560  001002                    BNE     QQQ25           ;BRANCH IF INCORRECT.
20652  105562  077203                    SOB     R2,1$
20653  105564  000401                    BR      QQQDONE
20654                              ;REPORT DATA INCORRECT.
20655  105566                      QQQ25:
  (2)  105566  104000                    EMT                     ;
20656  105600                      QQQDONE:
  (1)  105600  004767  016746            JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                            ;SEE IF THE USER HAS EXPRESSED
  (1)                                                            ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                            ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                            ;THE USER TYPED CONTROL G?).
20657
20663                              ;*****************************************************************************
  (2)                              ;TEST 515       FDST MODE 2, WITH GR7, TEST
  (3)                              ;*****************************************************************************
  (2)  105604                      TS515:
20664
20665  105604  012700  105654            MOV     #RRR3,R0        ;SET UP THE DATA BUFFER FOLLOWING THE TEST INSTRUCTION.
20666  105610  012701  105722            MOV     #RRRTP1,R1
20667  105614  012702  000004            MOV     #4,R2
20668  105620  012021                1$: MOV     (R0)+,(R1)+
20669  105622  077202                    SOB     R2,1$
20670  105624  012700  000200            MOV     #200,R0         ;ENTER FLOATING DOUBLE MODE.
20671  105630  170100                    LDFPS   R0
20672  105632  012700  105732            MOV     #RRRTP2,R0      ;SET UP AC0.
20673  105636  172410                    LDD     (R0),AC0
20674  105640  012737  105720  000004    MOV     #RRR10,@#ERRVECT ;SET UP FOR AN ODD ADDRESS.
20675  105646  005001                    CLR     R1
20676  105650  005004                    CLR     R4
20677                              ;THIS IS THE TEST INSTRUCTION. IT SHOULD MODIFY THE FIRST LOCATION
20678                              ;AFTER IT TO BE AN INCREMENT R4, INC R4, INSTRUCTION INSTEAD
20679                              ;OF AN INCREMENT R1 INSTRUCTION. THE INCREMENT R4 SHOULD NOT BE
20680                              ;EXECUTED SINCE THE PC SHOULD BE INCREMENTED BY TWO DURING IMMEDIATE
20681                              ;MODE ADDRESSING. THUS AFTER THE EXECUTION OF THE NEXT 5 INSTRUCTIONS
20682                              ;R1 SHOULD CONTAIN 3 AND R4 SHOULD CONTAIN 0.
20683  105652  174027              RRR2:   STD     AC0,(R7)+       ;TEST INSTRUCTION.
20684  105654  005201              RRR3:   INC     R1              ;THE STD INSTRUCTION SHOULD CHANGE THIS TO INC R4.
20685  105656  005201                      INC     R1
20686  105660  005201                      INC     R1
20687  105662  005201                      INC     R1
20688  105664  012700  105742            MOV     #RRREXP,R0      ;SEE IF THE DATA WAS OUTPUT CORRECTLY.
20689  105670  012702  105654            MOV     #RRR3,R2
20690  105674  012703  000004            MOV     #4,R3
20691  105700  022022              RRR4:   CMP     (R0)+,(R2)+
20692  105702  001006                      BNE     RRR10           ;BRANCH IF INCORRECT.
20693  105704  077303                      SOB     R3,RRR4
20694  105706  005704                      TST     R4              ;MAKE SURE R4 IS 0.
20695  105710  001003                      BNE     RRR10           ;BRANCH IF R4 IS INCORRECT.
20696  105712  022701  000003            CMP     #3,R1           ;SEE IF R1 IS CORRECT.
20697  105716  001415                    BEQ     RRRDONE
  (1)  105720                      RRR10:
  (2)  105720  104000                    EMT                     ;
20698                              ;THESE ARE TEST DATA PATTERNS USED TO SET UP THE OUTPUT BUFFER AT RRR3.
```

```
20699  105722  005201                  RRRTP1: INC     R1
20700  105724  005201                          INC     R1
20701  105726  005201                          INC     R1
20702  105730  005201                          INC     R1
20703                                  ;THIS IS THE DATA PUT IN AC0 BEFORE EXECUTION OF THE STD.
20704  105732  005204                  RRRTP2: INC     R4
20705  105734  005204                          INC     R4
20706  105736  005204                          INC     R4
20707  105740  005204                          INC     R4
20708                                  ;THIS IS THE EXPECTED DATA AT RRR3 AFTER EXECUTION OF THE STD.
20709  105742  005204                  RRREXP: INC     R4
20710  105744  005201                          INC     R1
20711  105746  005201                          INC     R1
20712  105750  005201                          INC     R1
20713  105752                          RRRDONE:
  (1)  105752  004767  016574                JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                ;THE USER TYPED CONTROL G?).
20714
20720                                  ;*****************************************************************************
  (2)                                  ;TEST 516       FDST MODE 4 TEST
  (3)                                  ;*****************************************************************************
  (2)  105756                          TS516:
20721
20722  105756  012700  177777                  MOV     #-1,R0          ;SET UP THE OUTPUT BUFFER.
20723  105762  012701  106104                  MOV     #SSSBF0,R1
20724  105766  012702  000010                  MOV     #10,R2
20725  105772  010021                  1$:     MOV     R0,(R1)+
20726  105774  077202                          SOB     R2,1$
20727  105776  012700  000200                  MOV     #200,R0         ;ENTER FLOATING DOUBLE MODE.
20728  106002  170100                          LDFPS   R0
20729  106004  012700  106124                  MOV     #SSSTP1,R0      ;SET UP AC0.
20730  106010  172410                          LDD     (R0),AC0
20731  106012  012737  106134  000004          MOV     #SSS10,@#ERRVECT ;SET UP FOR A TRAP TO 4.
20732  106020  012700  106114                  MOV     #SSSA1,R0       ;SET UP THE DESTINATION ADDRESS.
20733
20734  106024  174040                  SSS2:   STD     AC0,-(R0)       ;TEST INSTRUCTION.
20735  106026  005201                          INC     R1
20736  106030  020027  106104                  CMP     R0,#SSSBF0      ;SEE IF R0 WAS DECREMENTED PROPERLY.
20737  106034  001037                          BNE     SSS10           ;BRANCH IF R0 IS INCORRECT.
20738  106036  012700  106104                  MOV     #SSSBF0,R0      ;WAS THE OUTPUT DATA CORRECT?
20739  106042  012701  106124                  MOV     #SSSTP1,R1
20740  106046  012702  000004                  MOV     #4,R2
20741  106052  022021                  1$:     CMP     (R0)+,(R1)+
20742  106054  001027                          BNE     SSS10           ;BRANCH IF INCORRECT.
20743  106056  077203                          SOB     R2,1$
20744  106060  012700  177777                  MOV     #-1,R0          ;IS THE REST OF THE OUTPUT BUFFER CORRECT, -1?
20745  106064  012701  106114                  MOV     #SSSA1,R1
20746  106070  012702  000004                  MOV     #4,R2
20747  106074  020021                  2$:     CMP     R0,(R1)+
20748  106076  001016                          BNE     SSS10           ;BRANCH IF INCORRECT.
20749  106100  077203                          SOB     R2,2$
20750  106102  000415                          BR      SSSDONE
20751
```

```
20752                                   ;THIS IS THE OUTPUT DATA BUFFER.
20753  106104  177777                   SSSBF0: -1
20754  106106  177777                           -1
20755  106110  177777                           -1
20756  106112  177777                           -1
20757  106114  177777                   SSSA1:  -1
20758  106116  177777                           -1
20759  106120  177777                           -1
20760  106122  177777                           -1
20761
20762                                   ;THIS IS THE TEST DATA LOADED INTO AC0:
20763  106124  147250                   SSSTP1: 147250
20764  106126  036147                           36147
20765  106130  025036                           25036
20766  106132  147250                           147250
20767
20768  106134                           SSS10:
  (2)  106134  104000                           EMT                     ;
20769  106136                           SSSDONE:
  (1)  106136  004767  016410                   JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                   ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                   ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                   ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                   ;THE USER TYPED CONTROL G?).
20770
20776                                   ;*****************************************************************************
  (2)                                   ;TEST 517       FDST MODE 3 TEST
  (3)                                   ;*****************************************************************************
  (2)  106142                           TS517:
20777
20778  106142  012701  106252                   MOV     #TTTBF0,R1      ;SET UP THE OUTPUT DATA BUFFER.
20779  106146  012700  177777                   MOV     #-1,R0
20780  106152  012702  000013                   MOV     #13,R2
20781  106156  010021                   1$:     MOV     R0,(R1)+
20782  106160  077202                           SOB     R2,1$
20783  106162  012737  106252  106266           MOV     #TTTBF0,@#TTTA2
20784  106170  012700  000200                   MOV     #200,R0         ;ENTER DOUBLE FLOATING MODE.
20785  106174  170100                           LDFPS   R0
20786  106176  012700  106270                   MOV     #TTTTP1,R0      ;SET UP AC0.
20787  106202  172410                           LDD     (R0),AC0
20788  106204  012737  106300  000004           MOV     #TTT10,@#ERRVECT ;SET UP FOR TRAPS TO 4.
20789  106212  012700  106266                   MOV     #TTTA2,R0       ;SET UP THE DESTINATION ADDRESS.
20790
20791  106216  174030                   TTT2:   STD     AC0,@(R0)+      ;TEST INSTRUCTION.
20792
20793  106220  020027  106270                   CMP     R0,#TTTA2+2     ;SEE IF R0 WAS INCREMENTED CORRECTLY.
20794  106224  001025                           BNE     TTT10           ;BRANCH IF INCORRECT.
20795  106226  012701  106252                   MOV     #TTTBF0,R1      ;CHECK THE OUTPUT DATA BUFFER.
20796  106232  012702  106270                   MOV     #TTTTP1,R2
20797  106236  012703  000004                   MOV     #4,R3
20798  106242  022122                   TTT3:   CMP     (R1)+,(R2)+
20799  106244  001015                           BNE     TTT10           ;BRANCH IF NOT CORRECT.
20800  106246  077303                           SOB     R3,TTT3
20801  106250  000414                           BR      TTTDONE
20802
20803                                   ;THIS IS THHE OUTPUT DATA BUFFER:
```

```
20804  106252  177777                  TTTBF0: -1
20805  106254  177777                          -1
20806  106256  177777                          -1
20807  106260  177777                          -1
20808  106262  177777                          -1
20809  106264  177777                  TTTA1:  -1
20810  106266  106252                  TTTA2:  TTTBF0
20811  106270  101213                  TTTTP1: 101213
20812  106272  141516                          141516
20813  106274  071727                          71727
20814  106276  037475                          37475
20815
20816  106300                          TTT10:
  (2)  106300  104000                          EMT                     ;
20817
20818  106302                          TTTDONE:
  (1)  106302  004767  016244                  JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
20819
20825                             ;*************************************************************************************
  (2)                             ;TEST 520       FDST MODE 5 TEST
  (3)                             ;*************************************************************************************
  (2)  106306                     TS520:
20826
20827  106306  012701  106416                  MOV     #UUUBF0,R1      ;SET UP THE OUTPUT DATA BUFFER.
20828  106312  012700  177777                  MOV     #-1,R0
20829  106316  012702  000013                  MOV     #13,R2
20830  106322  010021                  1$:     MOV     R0,(R1)+
20831  106324  077202                          SOB     R2,1$
20832  106326  012737  106416  106430          MOV     #UUUBF0,@#UUUA1
20833  106334  012700  000200                  MOV     #200,R0         ;ENTER DOUBLE FLOATING MODE.
20834  106340  170100                          LDFPS   R0
20835  106342  012700  106434                  MOV     #UUUTP1,R0      ;SET UP AC0.
20836  106346  172410                          LDD     (R0),AC0
20837  106350  012737  106444  000004          MOV     #UUU10,@#ERRVECT ;GET READY FOR ANY TRAPS TO 4.
20838  106356  012700  106432                  MOV     #UUUA2,R0       ;SET UP THE DESTINATION ADDRESS.
20839  106362  174050                  UUU2:   STD     AC0,@-(R0)      ;TEST INSTRUCTION.
20840  106364  020027  106430                  CMP     R0,#UUUA2-2     ;WAS R0 DECRIMENTED PROPERLY?
20841  106370  001025                          BNE     UUU10           ;BRANCH IF R0 IS INCORRECT.
20842  106372  012701  106416                  MOV     #UUUBF0,R1      ;WAS THE DATA OUTPUT CORRECTLY?
20843  106376  012702  106434                  MOV     #UUUTP1,R2
20844  106402  012703  000004                  MOV     #4,R3
20845  106406  022122                  UUU3:   CMP     (R1)+,(R2)+
20846  106410  001015                          BNE     UUU10           ;BRANCH IF DATA IS INCORRECT.
20847  106412  077303                          SOB     R3,UUU3
20848  106414  000414                          BR      UUUDONE
20849
20850                                   ;THIS IS THE OUTPUT DATA BUFFER
20851  106416  177777                  UUUBF0: -1
20852  106420  177777                          -1
20853  106422  177777                          -1
20854  106424  177777                          -1
20855  106426  177777                          -1
```

```
20856  106430  106416                  UUUA1:  UUUBF0
20857  106432  177777                  UUUA2:  -1
20858  106434  020212                  UUUTP1: 20212
20859  106436  023242                          23242
20860  106440  026273                          26273
20861  106442  031323                          031323
20862
20863  106444                          UUU10:
  (2)  106444  104000                          EMT                     ;
20864  106446                          UUUDONE:
  (1)  106446  004767  016100                  JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
20865
20871                             ;*************************************************************************************
  (2)                             ;TEST 521       FDST MODE 6, INDEX MODE, TEST
  (3)                             ;*************************************************************************************
  (2)  106452                     TS521:
20872
20873  106452  012700  000200                  MOV     #200,R0         ;ENTER DOUBLE FLOATING MODE.
20874  106456  170100                          LDFPS   R0
20875  106460  012701  106562                  MOV     #VVVBF0,R1      ;SET UP THE OUT PUT DATA BUFFER.
20876  106464  012700  177777                  MOV     #-1,R0
20877  106470  012702  000004                  MOV     #4,R2
20878  106474  010021                  1$:     MOV     R0,(R1)+
20879  106476  077202                          SOB     R2,1$
20880  106500  012737  106602  000004          MOV     #VVV10,@#ERRVECT ;SET UP VECTOR 4 INCASE OF ERROR.
20881  106506  012700  106572                  MOV     #VVVTP1,R0      ;SET UP AC0.
20882  106512  172410                          LDD     (R0),AC0
20883  106514  012700  100661                  MOV     #VVVBF0-5701,R0 ;SET UP THE DESTINATION ADDRESS.
20884  106520  012701  000001                  MOV     #1,R1
20885  106524  174060  005701          VVV2:   STD     AC0,5701(R0)    ;TEST INSTRUCTION.
20886
20887  106530  020027  100661                  CMP     R0,#VVVBF0-5701 ;SEE IF R0 WAS MODIFIED.
20888  106534  001022                          BNE     VVV10           ;BRANCH IF INCORRECT.
20889  106536  012702  106562                  MOV     #VVVBF0,R2      ;WAS THE OUTPUT DATA CORRECT.
20890  106542  012703  106572                  MOV     #VVVTP1,R3
20891  106546  012704  000004                  MOV     #4,R4
20892  106552  022223                  1$:     CMP     (R2)+,(R3)+
20893  106554  001012                          BNE     VVV10           ;BRANCH IF INCORRECT DATA.
20894  106556  077403                          SOB     R4,1$
20895  106560  000411                          BR      VVVDONE
20896  106562  177777                  VVVBF0: -1
20897  106564  177777                          -1
20898  106566  177777                          -1
20899  106570  177777                          -1
20900  106572  030313                  VVVTP1: 30313
20901  106574  023334                          23334
20902  106576  035363                          35363
20903  106600  074041                          74041
20904
20905  106602                          VVV10:
  (2)  106602  104000                          EMT                     ;
20906  106604                          VVVDONE:
```

```
  (1)  106604  004767  015742                    JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                      ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                      ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                      ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                      ;THE USER TYPED CONTROL G?).
20907
20913                                ;*************************************************************************
  (2)                                ;TEST 522       FDST MODE 7, INDEX DEFERRED MODE, TEST
  (3)                                ;*************************************************************************
  (2)  106610                        TS522:
20914
20915  106610  012700  000200                    MOV     #200,R0         ;ENTER DOUBLE FLOATING MODE.
20916  106614  170100                            LDFPS   R0
20917  106616  012701  106726                    MOV     #WWWBF0,R1      ;SET UP THE OUTPUT DATA BUFFER.
20918  106622  012700  177777                    MOV     #-1,R0
20919  106626  012702  000004                    MOV     #4,R2
20920  106632  010021                    1$:     MOV     R0,(R1)+
20921  106634  077202                            SOB     R2,1$
20922  106636  012737  106756  000004           MOV     #WWW10,@#ERRVECT ;SET UP FOR TRAPS TO 4.
20923  106644  012700  106736                    MOV     #WWWTP1,R0      ;SET UP AC0.
20924  106650  172410                            LDD     (R0),AC0
20925  106652  012700  101045                    MOV     #WWWBF1-5701,R0 ;SET UP THE DESTINATION ADDRESS.
20926  106656  012701  000001                    MOV     #1,R1
20927  106662  012737  106726  106746           MOV     #WWWBF0,@#WWWBF1
20928  106670  174070  005701            WWW2:   STD     AC0,@5701(R0)   ;TEST INSTRUCTION.
20929
20930  106674  020027  101045                    CMP     R0,#WWWBF1-5701 ;IS R0 CORRECT?
20931  106700  001026                            BNE     WWW10           ;BRANCH IF INCORRECT.
20932  106702  012702  106726                    MOV     #WWWBF0,R2      ;WAS THE DATA OUTPUT CORRECTLY?
20933  106706  012703  106736                    MOV     #WWWTP1,R3
20934  106712  012704  000004                    MOV     #4,R4
20935  106716  022223                    1$:     CMP     (R2)+,(R3)+
20936  106720  001016                            BNE     WWW10           ;BRANCH IF DATA IS INCORRECT.
20937  106722  077403                            SOB     R4,1$
20938  106724  000415                            BR      WWWDONE
20939  106726  177777                    WWWBF0: -1
20940  106730  177777                            -1
20941  106732  177777                            -1
20942  106734  177777                            -1
20943  106736  041424                    WWWTP1: 41424
20944  106740  034445                            34445
20945  106742  046475                            46475
20946  106744  051525                            051525
20947  106746  177777                    WWWBF1: -1
20948  106750  177777                            -1
20949  106752  177777                            -1
20950  106754  177777                            -1
20951
20952  106756                            WWW10:
  (2)  106756  104000                            EMT                     ;
20953  106760                            WWWDONE:
  (1)  106760  004767  015566                    JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                      ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                      ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                      ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                      ;THE USER TYPED CONTROL G?).
```

```
20954
20960                                   ;*****************************************************************************
  (2)                                   ;TEST 523       STCFD TEST
  (3)                                   ;*****************************************************************************
  (2) 106764                            TS523:
20961
20962                                   ;AC=0
20963  106764  004767  000262           XXX1:   JSR     PC,STCFDS
20964  106770  000000                   1$:     0               ;AC
20965  106772  000000                           0
20966  106774  000000                           0
20967  106776  000000                           0
20968  107000  000000                   2$:     0               ;RES
20969  107002  000000                           0
20970  107004  000000                           0
20971  107006  000000                           0
20972  107010  000000                   3$:     0               ;ERROR RES.
20973  107012  000000                           0
20974  107014  177777                           -1
20975  107016  177777                           -1
20976  107020  047000                   4$:     47000           ;FPS BEFORE EXECUTION.
20977  107022  047004                           47004           ;FPS AFTER EXECUTION.
20978  107024  177777                           -1              ;FEC
20979  107026  147004                           147004          ;ERROR FPS.
20980
20981
20982  107030  004767  000216           XXX2:   JSR     PC,STCFDS
20983  107034  017203                   1$:     17203           ;AC
20984  107036  142536                           142536
20985  107040  047506                           47506
20986  107042  172031                           172031
20987  107044  017203                   2$:     17203           ;RES
20988  107046  142536                           142536
20989  107050  000000                           0
20990  107052  000000                           0
20991  107054  017203                   3$:     17203           ;ERROR RES.
20992  107056  142536                           142536
20993  107060  047506                           47506
20994  107062  172031                           172031
20995  107064  040000                   4$:     40000           ;FPS BEFORE EXECUTION.
20996  107066  040000                           40000           ;FPS AFTER EXECUTION.
20997  107070  177777                           -1              ;FEC
20998  107072  177777                           -1              ;ERROR FPS.
20999
21000
21001  107074  004767  000152           XXX3:   JSR     PC,STCFDS
21002  107100  050717                   1$:     50717           ;AC
21003  107102  027374                           27374
21004  107104  075767                           75767
21005  107106  077071                           77071
21006  107110  050717                   2$:     50717           ;RES
21007  107112  027374                           27374
21008  107114  000000                           0
21009  107116  000000                           0
21010  107120  000000                   3$:     0               ;ERROR RES.
21011  107122  000000                           0
```

```
21012  107124  000000                          0
21013  107126  000000                          0
21014  107130  047000                 4$:      47000          ;FPS BEFORE EXECUTION.
21015  107132  047000                          47000          ;FPS AFTER EXECUTION.
21016  107134  177777                          -1             ;FEC
21017  107136  174002                          174002         ;ERROR FPS.
21018
21019
21020  107140  004767  000106         XXX4:    JSR     PC,STCFDS
21021  107144  020212                 1$:      20212          ;AC
21022  107146  032425                          32425
21023  107150  026272                          26272
21024  107152  002123                          02123
21025  107154  020212                 2$:      20212          ;RES
21026  107156  032425                          32425
21027  107160  000000                          0
21028  107162  000000                          0
21029  107164  020212                 3$:      20212          ;ERROR RES.
21030  107166  032425                          32425
21031  107170  100000                          100000
21032  107172  000000                          0
21033  107174  040000                 4$:      40000          ;FPS BEFORE EXECUTION.
21034  107176  040000                          40000          ;FPS AFTER EXECUTION.
21035  107200  177777                          -1             ;FEC
21036  107202  177777                          -1             ;ERROR FPS.
21037
21038
21039  107204  004767  000042         XXX5:    JSR     PC,STCFDS
21040  107210  121314                 1$:      121314         ;AC
21041  107212  151617                          151617
21042  107214  101112                          101112
21043  107216  131415                          131415
21044  107220  121314                 2$:      121314         ;RES
21045  107222  151617                          151617
21046  107224  000000                          0
21047  107226  000000                          0
21048  107230  021314                 3$:      21314          ;ERROR RES.
21049  107232  151617                          151617
21050  107234  000000                          0
21051  107236  000000                          0
21052  107240  040000                 4$:      40000          ;FPS BEFORE EXECUTION.
21053  107242  040010                          40010          ;FPS AFTER EXECUTION.
21054  107244  177777                          -1             ;FEC
21055  107246  177777                          -1             ;ERROR FPS.
21056  107250  000460                 6$:      BR      XXXDONE
21057
21058                                 ;
21059
21060
21061                                 ;THIS SUBROUTINE, STCFDS, IS USED TO SET UP THE OPERANDS, EXECUTE
21062                                 ;THE STCFD INSTRUCTION AND CHECK THE RESULTS. A CALL
21063                                 ;TO IT IS MADE THUS:
21064                                 ;
21065                                 ;               JSR     PC,@#STCFDS
21066                                 ;               ACARG:  .WORD   X,X,X,X         ;AC OPERAND
21067                                 ;               RES:    .WORD   X,X,X,X         ;EXPECTED RESULT
```

```
21068                                   ;              ERRES:  .WORD   X,X,X,X          ;ERROR RESULT
21069                                   ;              FPSB:   .WORD   X                ;FPS BEFORE EXECUTION
21070                                   ;              FPSA:   .WORD   X                ;FPS AFTER EXECUTION
21071                                   ;              FEC:    .WORD   X                ;EXPECTED FEC
21072                                   ;              ERFPS:  .WORD   X                ;ERROR FPS.
21073                                   ;              ERR1:   ERROR   X                ;DATA ERROR.
21074                                   ;                      BR      CONT
21075                                   ;              ERR2:   ERROR   X                ;FPS ERROR.
21076                                   ;              CONT:                            ;RETURN ADDRESS
21077                                   ;
21078                                   ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
21079                                   ;THE STCFD INSTRUCTION IS EXECUTED.
21080                                   ;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
21081                                   ;COMPARED WITH FPSA IF THIS TOO IS CORRECT STCFDS RETURNS CONTROL
21082                                   ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD STCFDS
21083                                   ;COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN STCFDS WILL RETURN
21084                                   ;TO THE ERROR CALL AT ERR2, OTHERWISE STCFDS ITSELF
21085                                   ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
21086                                   ;STCFD IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
21087                                   ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
21088                                   ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN STCFDS
21089                                   ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
21090                                   ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND STCFDS WILL
21091                                   ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.
21092
21093  107252  012601                   STCFDS: MOV     (SP)+,R1                 ;PICK UP THE POINTER TO THE OPERANDS.
21094  107254  012700  000200                   MOV     #200,R0          ;ENTER DOUBLE FLOATING MODE.
21095  107260  170100                           LDFPS   R0
21096  107262  010100                           MOV     R1,R0            ;LOAD AC0.
21097  107264  172410                           LDD     (R0),AC0
21098  107266  012700  177777                   MOV     #-1,R0           ;FILL THE OUTPUT BUFFER WITH -1'S.
21099  107272  012702  107402                   MOV     #STCFT,R2
21100  107276  012703  000004                   MOV     #4,R3
21101  107302  010022                   1$:     MOV     R0,(R2)+
21102  107304  077302                           SOB     R3,1$
21103  107306  016100  000030                   MOV     30(R1),R0        ;LOAD THE FPS.
21104  107312  170100                           LDFPS   R0
21105  107314  012700  107402                   MOV     #STCFT,R0        ;SET UP THE DESTINATION ADDRESS.
21106  107320  176010                   2$:     STCFD   AC0,(R0)         ;TEST INSTRUCTION.
21107
21108  107322  170204                           STFPS   R4               ;GET THE FPS.
21109  107324  170305                           STST    R5               ;GET THE FEC.
21110  107326  010102                           MOV     R1,R2            ;CHECK THE RESULT.
21111  107330  062702  000010                   ADD     #10,R2
21112  107334  012703  107402                   MOV     #STCFT,R3
21113  107340  012700  000004                   MOV     #4,R0
21114  107344  022223                   3$:     CMP     (R2)+,(R3)+
21115  107346  001014                           BNE     10$              ;BRANCH IF INCORRECT.
21116  107350  077003                           SOB     R0,3$
21117
21118  107352  016102  000032                   MOV     32(R1),R2
21119  107356  020204                           CMP     R2,R4            ;IS THE FPS CORRECT?
21120  107360  001007                           BNE     10$              ;BRANCH IF FPS INCORRECT.
21121  107362  005702                           TST     R2               ;IF EXPECTED FPS IS NEGATIVE, THEN
21122  107364  100003                           BPL     4$               ;GO AHEAD AND CHECK THE FEC.
21123  107366  026105  000036                   CMP     36(R1),R5
```

```
21124  107372  001002                          BNE     10$             ;BRANCH IF FEC IS INCORRECT.
21125  107374  000161  000040          4$:     JMP     40(R1)          ;RETURN.
21126  107400                          10$:
  (2)  107400  104000                          EMT                     ;
21127  107402  177777  177777  177777  STCFT:  -1,-1,-1,-1
       107410  177777
21128  107412                          XXXDONE:
  (1)  107412  004767  015134                  JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
21129
21135                                  ;*************************************************************************************
  (2)                                  ;TEST 524       STCDF TEST
  (3)                                  ;*************************************************************************************
  (2)  107416                          TS524:
21136
21137                                  ;AC=0
21138  107416  004767  000262          YYY1:   JSR     PC,STCDFS
21139  107422  000000                  1$:     0                       ;AC
21140  107424  000000                          0
21141  107426  000000                          0
21142  107430  000000                          0
21143  107432  000000                  2$:     0                       ;RES
21144  107434  000000                          0
21145  107436  177777                          -1
21146  107440  177777                          -1
21147  107442  000000                  3$:     0                       ;ERROR RES.
21148  107444  000000                          0
21149  107446  000000                          0
21150  107450  000000                          0
21151  107452  047200                  4$:     47200                   ;FPS BEFORE EXECUTION.
21152  107454  047204                          47204                   ;FPS AFTER EXECUTION.
21153  107456  177777                          -1                      ;FEC
21154  107460  177777                          -1                      ;ERROR FPS.
21155
21156
21157  107462  004767  000216          YYY2:   JSR     PC,STCDFS
21158  107466  067574                  1$:     67574                   ;AC0
21159  107470  073727                          73727
21160  107472  170777                          170777
21161  107474  067574                          67574
21162  107476  067574                  2$:     67574                   ;RES
21163  107500  073730                          73730
21164  107502  177777                          -1
21165  107504  177777                          -1
21166  107506  067574                  3$:     67574                   ;ERROR RES.
21167  107510  073727                          73727
21168  107512  177777                          -1
21169  107514  177777                          -1
21170  107516  040200                  4$:     40200                   ;FPS BEFORE EXECUTION.
21171  107520  040200                          40200                   ;FPS AFTER EXECUTION.
21172  107522  177777                          -1                      ;FEC
21173  107524  177777                          -1                      ;ERROR FPS.
21174
```

F13



```
21175
21176  107526  004767  000152          YYY3:   JSR     PC,STCDFS
21177  107532  077777                  1$:     77777                   ;AC0
21178  107534  177777                          -1
21179  107536  100000                          100000
21180  107540  000000                          0
21181  107542  000000                  2$:     0                       ;RES
21182  107544  000000                          0
21183  107546  177777                          -1
21184  107550  177777                          -1
21185  107552  077777                  3$:     77777                   ;ERROR RES.
21186  107554  177777                          -1
21187  107556  177777                          -1
21188  107560  177777                          -1
21189  107562  040200                  4$:     40200                   ;FPS BEFORE EXECUTION.
21190  107564  040206                          40206                   ;FPS AFTER EXECUTION.
21191  107566  177777                          -1                      ;FEC
21192  107570  040204                          40204                   ;ERROR FPS.
21193
21194
21195  107572  004767  000106          YYY4:   JSR     PC,STCDFS
21196  107576  077777                  1$:     77777                   ;AC0
21197  107600  177777                          -1
21198  107602  100000                          100000
21199  107604  000000                          0
21200  107606  000000                  2$:     0                       ;RES
21201  107610  000000                          0
21202  107612  177777                          -1
21203  107614  177777                          -1
21204  107616  077777                  3$:     77777                   ;ERROR RES.
21205  107620  177777                          -1
21206  107622  177777                          -1
21207  107624  177777                          -1
21208  107626  040200                  4$:     40200                   ;FPS BEFORE EXECUTION.
21209  107630  040206                          40206                   ;FPS AFTER EXECUTION.
21210  107632  177777                          -1                      ;FEC
21211  107634  140206                          140206                  ;ERROR FPS.
21212
21213
21214  107636  004767  000042          YYY5:   JSR     PC,STCDFS
21215  107642  177777                  1$:     177777                  ;AC0
21216  107644  177777                          -1
21217  107646  100000                          100000
21218  107650  000000                          0
21219  107652  100000                  2$:     100000                  ;RES
21220  107654  000000                          0
21221  107656  177777                          -1
21222  107660  177777                          -1
21223  107662  000000                  3$:     0                       ;ERROR RES.
21224  107664  000000                          0
21225  107666  177777                          -1
21226  107670  177777                          -1
21227  107672  047200                  4$:     47200                   ;FPS BEFORE EXECUTION.
21228  107674  147216                          147216                  ;FPS AFTER EXECUTION.
21229  107676  000010                          10                      ;FEC
21230  107700  047206                          47206                   ;ERROR FPS.
```

```
21231  107702  000460                    6$:     BR      YYYDONE
21232                                    ;THIS SUBROUTINE, STCDFS, IS USED TO SET UP THE OPERANDS, EXECUTE
21233                                    ;THE STCDF INSTRUCTION AND CHECK THE RESULTS. A CALL
21234                                    ;TO IT IS MADE THUS:
21235                                    ;
21236                                    ;               JSR     PC,@#STCFDS
21237                                    ;               ACARG:  .WORD   X,X,X,X         ;AC OPERAND
21238                                    ;               RES:    .WORD   X,X,X,X         ;EXPECTED RESULT
21239                                    ;               ERRES:  .WORD   X,X,X,X         ;ERROR RESULT
21240                                    ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
21241                                    ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
21242                                    ;               FEC:    .WORD   X               ;EXPECTED FEC
21243                                    ;               ERFPS:  .WORD   X               ;ERROR FPS.
21244                                    ;               ERR1:   ERROR   X               ;DATA ERROR.
21245                                    ;                       BR      CONT
21246                                    ;               ERR2:   ERROR   X               ;FPS ERROR.
21247                                    ;               CONT:                           ;RETURN ADDRESS
21248                                    ;
21249                                    ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
21250                                    ;THE STCFD INSTRUCTION IS EXECUTED.
21251                                    ;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
21252                                    ;COMPARED WITH FPSA IF THIS TOO IS CORRECT STCFDS RETURNS CONTROL
21253                                    ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD STCFDS
21254                                    ;COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN STCFDS WILL RETURN
21255                                    ;TO THE ERROR CALL AT ERR2, OTHERWISE STCFDS ITSELF
21256                                    ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
21257                                    ;STCFD IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
21258                                    ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
21259                                    ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN STCFDS
21260                                    ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
21261                                    ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND STCFDS WILL
21262                                    ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.
21263
21264  107704  012601                    STCDFS: MOV     (SP)+,R1        ;PICK UP THE POINTER TO THE OPERANDS.
21265  107706  012700  000200                    MOV     #200,R0         ;ENTER DOUBLE FLOATING MODE.
21266  107712  170100                            LDFPS   R0
21267  107714  010100                            MOV     R1,R0           ;LOAD AC0.
21268  107716  172410                            LDD     (R0),AC0
21269  107720  012700  177777                    MOV     #-1,R0          ;FILL THE OUTPUT BUFFER WITH -1'S.
21270  107724  012702  110034                    MOV     #STCDT,R2
21271  107730  012703  000004                    MOV     #4,R3
21272  107734  010022                    1$:     MOV     R0,(R2)+
21273  107736  077302                            SOB     R3,1$
21274  107740  016100  000030                    MOV     30(R1),R0       ;LOAD THE FPS.
21275  107744  170100                            LDFPS   R0
21276  107746  012700  110034                    MOV     #STCDT,R0       ;SET UP THE DESTINATION ADDRESS.
21277  107752  176010                    2$:     STCDF   AC0,(R0)        ;TEST INSTRUCTION.
21278
21279  107754  170204                            STFPS   R4              ;GET THE FPS.
21280  107756  170305                            STST    R5              ;GET THE FEC.
21281  107760  010102                            MOV     R1,R2           ;CHECK THE RESULT.
21282  107762  062702  000010                    ADD     #10,R2
21283  107766  012703  110034                    MOV     #STCDT,R3
21284  107772  012700  000004                    MOV     #4,R0
21285  107776  022223                    3$:     CMP     (R2)+,(R3)+
21286  110000  001014                            BNE     10$             ;BRANCH IF INCORRECT.
```

```
21287  110002  077003                        SOB     R0,3$
21288
21289  110004  016102  000032                MOV     32(R1),R2
21290  110010  020204                        CMP     R2,R4           ;IS THE FPS CORRECT?
21291  110012  001007                        BNE     10$             ;BRANCH IF FPS INCORRECT.
21292  110014  005702                        TST     R2              ;IF EXPECTED FPS IS NEGATIVE, THEN
21293  110016  100003                        BPL     4$              ;GO AHEAD AND CHECK THE FEC.
21294  110020  026105  000034                CMP     34(R1),R5
21295  110024  001002                        BNE     10$             ;BRANCH IF FEC IS INCORRECT.
21296  110026  000161  000040        4$:     JMP     40(R1)          ;RETURN.
21297  110032                        10$:
  (2)  110032  104000                        EMT                     ;
21298  110034  177777  177777  177777  STCDT:  -1,-1,-1,-1
       110042  177777
21299  110044                        YYYDONE:
  (1)  110044  004767  014502                JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                ;THE USER TYPED CONTROL G?).
21305                           ;*************************************************************************************
  (2)                           ;TEST 525       STCFD WITH ILLEGAL ACCUMULATOR TEST
  (3)                           ;*************************************************************************************
  (2)  110050                   TS525:
21306
21307  110050  012700  040000                MOV     #40000,R0       ;DISSABLE INTERRUPTS.
21308  110054  170100                        LDFPS   R0
21309  110056  176006                ZZZ2:   STCFD   AC0,AC6         ;THIS TEST INSTRUCTION SHOULD CAUSE AN ERROR.
21310
21311  110060  170204                        STFPS   R4              ;GET FPS.
21312  110062  170305                        STST    R5              ;GET FEC.
21313  110064  020427  140000                CMP     R4,#140000      ;IS FPS CORRECT?
21314  110070  001004                        BNE     ZZZ10           ;BRANCH IF INCORRECT FPS.
21315  110072  022705  000002                CMP     #2,R5           ;IS FEC CORRECT?
21316  110076  001001                        BNE     ZZZ10           ;BRANCH IF INCORRECT.
21317  110100  000401                        BR      ZZZDONE
21318
21319  110102                        ZZZ10:
  (2)  110102  104000                        EMT                     ;
21320
21321  110104                        ZZZDONE:
  (1)  110104  004767  014442                JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                ;THE USER TYPED CONTROL G?).
21322
21328                           ;*************************************************************************************
  (2)                           ;TEST 526       CLRD TEST
  (3)                           ;*************************************************************************************
  (2)  110110                   TS526:
21329  110110  012700  110214                MOV     #AABTP1,R0      ;SET UP OUTPUT BUFFER
21330  110114  012701  110204                MOV     #AABBF0,R1
21331  110120  012702  000004                MOV     #4,R2
21332  110124  012021                1$:     MOV     (R0)+,(R1)+
21333  110126  077202                        SOB     R2,1$
```

```
21334  110130  012700  110204          MOV     #AABBF0,R0      ;SET UP DESTINATION OPERAND ADDRESS.
21335  110134  012701  000213          MOV     #213,R1         ;SET UP FPS.
21336  110140  170101                  LDFPS   R1
21337  110142  170410          2$:     CLRD    (R0)            ;TEST INSTRUCTION.
21338
21339  110144  170205                  STFPS   R5              ;GET FPS.
21340  110146  012702  000004          MOV     #4,R2           ;SEE IF RESULT CLEAR, 0.
21341  110152  012701  110204          MOV     #AABBF0,R1
21342  110156  005721          3$:     TST     (R1)+
21343  110160  001010                  BNE     AAB2            ;BRANCH IF RESULT INCORRECT, NOT 0.
21344  110162  077203                  SOB     R2,3$
21345  110164  022705  000204          CMP     #204,R5         ;SEE IF FPS IS CORRECT.
21346  110170  001004                  BNE     AAB2            ;BRANCH IF INCORRECT.
21347  110172  020027  110204          CMP     R0,#AABBF0      ;SEE IF R0 IS CORRECT.
21348  110176  001001                  BNE     AAB2            ;BRANCH IF R0 IS INCORRECT.
21349  110200  000411                  BR      AABDONE
21350
21351
21352  110202                  AAB2:
  (2)  110202  104000                  EMT                     ;
21353
21354                          ;THIS IS THE TEST DATA BUFFER, OUTPUT DATA BUFFER.
21355  110204  073475          AABBF0: 73475
21356  110206  067707                  67707
21357  110210  127347                  127347
21358  110212  056770                  56770
21359                          ;THIS IS THE DATA USED TO SET UP THE OUTPUT BUFFER.
21360  110214  073475          AABTP1: 73475
21361  110216  067707                  67707
21362  110220  127347                  127347
21363  110222  056770                  56770
21364  110224                  AABDONE:
  (1)  110224  004767  014322          JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                          ;SEE IF THE USER HAS EXPRESSED
  (1)                                                          ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                          ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                          ;THE USER TYPED CONTROL G?).
21365
21371                          ;*****************************************************************************
  (2)                          ;TEST 527       CLRD WITH ILLEGAL ACCUMULATOR TEST
  (3)                          ;*****************************************************************************
  (2)  110230             TS527:
21372  110230  012700  040200          MOV     #40200,R0       ;SET UP THE FPS, NO INTERRUPTS AND FD=1.
21373  110234  170100                  LDFPS   R0
21374  110236  170407          CCB2:   CLRD    AC7             ;TEST INSTRUCTION.
21375
21376  110240  170204                  STFPS   R4              ;GET FPS.
21377  110242  170305                  STST    R5              ;GET FEC.
21378  110244  020427  140200          CMP     R4,#140200              ;IS THE FPS CORRECT?
21379  110250  001004                  BNE     CCB10           ;BRANCH IF FPS IS INCORRECT.
21380  110252  022705  000002          CMP     #2,R5           ;IS THE FEC CORRECT?
21381  110256  001001                  BNE     CCB10           ;BRANCH IF FEC IS INCORRECT.
21382  110260  000401                  BR      CCBDONE
21383
21384  110262                  CCB10:
  (2)  110262  104000                  EMT                     ;
```

```
21385  110264
  (1)  110264  004767  014262          CCBDONE:
                                              JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                 ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                 ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                 ;THE USER TYPED CONTROL G?).
21394
21395                                 ;*************************************************************************************
  (2)                                 ;TEST 530       NEGF, ABSF AND TSTF SOURCE MODE 0 WITH ILLEGAL AC7, TEST
  (3)                                 ;*************************************************************************************
  (2)  110270                         TS530:
21396
21397  110270  012700  040200                 MOV     #40200,R0               ;SET UP THE FPS, FID=1 AND FD=1.
21398  110274  170100                         LDFPS   R0
21399  110276  170707                 VVB2:   NEGD    AC7             ;TEST INSTRUCTION.
21400
21401  110300  170204                         STFPS   R4              ;GET FPS.
21402  110302  170305                         STST    R5              ;GET FEC.
21403
21404  110304  022704  140200                 CMP     #140200,R4              ;IS FPS CORRECT?
21405  110310  001004                         BNE     VVB10           ;BRANCH IF FPS IS INCORRECT.
21406  110312  022705  000002                 CMP     #2,R5           ;IS FEC CORRECT?
21407  110316  001001                         BNE     VVB10           ;BRANCH IF FEC IS INCORRECT.
21408  110320  000401                         BR      VVBDONE
21409  110322                          VVB10:
  (2)  110322  104000                         EMT                     ;
21410
21411  110324                          VVBDONE:
  (1)  110324  004767  014222                JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                 ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                 ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                 ;THE USER TYPED CONTROL G?).
21412
21420                                 ;*************************************************************************************
  (2)                                 ;TEST 531       NEGF, ABSF AND TSTF SOURCE MODE 0 TEST
  (3)                                 ;*************************************************************************************
  (2)  110330                         TS531:
21421
21422  110330  012700  000200                 MOV     #200,R0         ;SET FD MODE.
21423  110334  170100                         LDFPS   R0
21424  110336  012700  110430                 MOV     #DD8TP1,R0      ;SET UP AC0.
21425  110342  172410                         LDD     (R0),AC0                ;SET AC0 = 0
21426  110344  005000                         CLR     R0              ;CLEAR THE FPS.
21427  110346  170100                         LDFPS   R0
21428  110350  012700  110440                 MOV     #DD8TP2,R0              ;LOAD AC0 TO BE A FLOATING 0.
21429  110354  172410                         LDF     (R0),AC0        ;SET AC0=ZERO
21430                                                                 ;FLOAT
21431  110356  012700  000201                 MOV     #201,R0         ;SET FD MODE.
21432  110362  170100                         LDFPS   R0
21433  110364  170700                 DD82:   NEGD    AC0             ;TEST INSTRUCTION
21434
21435  110366  170205                         STFPS   R5              ;GET FPS.
21436  110370  012700  000200                 MOV     #200,R0         ;SET FD MODE.
21437  110374  170100                         LDFPS   R0
21438  110376  012700  110450                 MOV     #DD8BF0,R0      ;GET THE RESULT OUT OF AC0.
```

```
21439  110402  174010                          STD     AC0,(R0)
21440                                                   ;SEE IF THE RESULT IS CORRECT.
21441  110404  012701  000004                  MOV     #4,R1
21442  110410  005720                  1$:     TST     (R0)+
21443  110412  001005                          BNE     DDB5            ;BRANCH IF THE RESULT IS INCORRECT.
21444  110414  077103                          SOB     R1,1$
21445  110416  022705  000204                  CMP     #204,R5         ;IS THE FPS CORRECT?
21446  110422  001001                          BNE     DDB5            ;BRANCH IF THE FPS IS INCORRECT.
21447  110424  000415                          BR      DDBDONE
21448  110426                          DDB5:
  (2)  110426  104000                          EMT                     ;
21449
21450                                   ;THESE ARE TEST DATA TABLES AND AN OUTPUT BUFFER.
21451  110430  101112                  DDBTP1: 101112
21452  110432  131415                          131415
21453  110434  161710                          161710
21454  110436  111213                          111213
21455  110440  000000                  DDBTP2: 0
21456  110442  000000                          0
21457  110444  000000                          0
21458  110446  000000                          0
21459
21460  110450  177777                  DDBBF0: -1
21461  110452  177777                          -1
21462  110454  177777                          -1
21463  110456  177777                          -1
21464
21465  110460                          DDBDONE:
  (1)  110460  004767  014066                  JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                   ;SEE IF THE USER HAS EXPRESSED
  (1)                                                   ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                   ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                   ;THE USER TYPED CONTROL G?).
21466
21467                                   ;*****************************************************************************
  (2)                                   ;TEST 532       NEGF, ABSF AND TSTF SOURCE MODE 1 TEST
  (3)                                   ;*****************************************************************************
  (2)  110464                           TS532:
21468
21469  110464  012700  110564                  MOV     #EEBTP1,R0      ;SET UP THE DATA BUFFER.
21470  110470  012701  110604                  MOV     #EEBBF1,R1
21471  110474  012702  000004                  MOV     #4,R2
21472  110500  012021                  1$:     MOV     (R0)+,(R1)+
21473  110502  077202                          SOB     R2,1$
21474  110504  012700  000200                  MOV     #200,R0         ;SET FD MODE.
21475  110510  170100                          LDFPS   R0
21476  110512  012700  110604                  MOV     #EEBBF1,R0      ;SET UP THE OPERAND ADDRESS.
21477  110516  012737  110614  000004          MOV     #EEB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF ERROR.
21478  110524  170710                  EEB2:   NEGD    (R0)            ;TEST INSTRUCTION.
21479
21480  110526  170205                          STFPS   R5              ;GET FPS.
21481  110530  012701  110604                  MOV     #EEBBF1,R1      ;SEE IF RESULT IS CORRECT.
21482  110534  012702  000004                  MOV     #4,R2
21483  110540  005721                  1$:     TST     (R1)+
21484  110542  001024                          BNE     EEB10           ;BRANCH IF NOT CORRECT.
21485  110544  077203                          SOB     R2,1$
```

```
21486
21487  110546  020027  110604              CMP     R0,#EEBBF1      ;IS R0 CORRECT?
21488  110552  001020                      BNE     EEB10                   ;BRANCH IF NOT CORRECT.
21489  110554  022705  000204              CMP     #204,R5         ;IS THE FPS CORRECT?
21490  110560  001015                      BNE     EEB10           ;BRANCH IF NOT CORRECT.
21491  110562  000415                      BR      EEBDONE
21492
21493                          ;THESE ARE TEST DATA TABLES AND A BUFFER.
21494  110564  000177          EEBTP1: 177
21495  110566  167574                  167574
21496  110570  137271                  137271
21497  110572  107675                  107675
21498  110574  177777          EEBBF0: -1
21499  110576  177777                  -1
21500  110600  177777                  -1
21501  110602  177777                  -1
21502  110604  177777          EEBBF1: -1
21503  110606  177777                  -1
21504  110610  177777                  -1
21505  110612  177777                  -1
21506  110614                  EEB10:
  (2)  110614  104000                  EMT                     ;
21507  110616                  EEBDONE:
  (1)  110616  004767  013730          JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                          ;SEE IF THE USER HAS EXPRESSED
  (1)                                                          ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                          ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                          ;THE USER TYPED CONTROL G?).
21508
21509                          ;*************************************************************************************
  (2)                          ;TEST 533       NEGF, ABSF AND TSTF SOURCE MODE 2 TEST
  (3)                          ;*************************************************************************************
  (2)  110622                  TS533:
21510
21511  110622  012700  110722          MOV     #FFBTP1,R0              ;SET UP THE DATA BUFFER.
21512  110626  012701  110732          MOV     #FFBBF1,R1
21513  110632  012702  000004          MOV     #4,R2
21514  110636  012021          1$:     MOV     (R0)+,(R1)+
21515  110640  077202                  SOB     R2,1$
21516  110642  012700  000200          MOV     #200,R0         ;SET FD.
21517  110646  170100                  LDFPS   R0
21518  110650  012700  110732          MOV     #FFBBF1,R0      ;SET UP THE OPERAND ADDRESS.
21519  110654  012737  110742  000004  MOV     #FFB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.
21520
21521  110662  170620          FFB2:   ABSD    (R0)+           ;TEST INSTRUCTION.
21522
21523  110664  170205                  STFPS   R5              ;GET FPS.
21524  110666  012701  110732          MOV     #FFBBF1,R1              ;CHECK RESULT.
21525  110672  012702  000004          MOV     #4,R2
21526  110676  005721          1$:     TST     (R1)+
21527  110700  001020                  BNE     FFB10           ;BRANCH IF INCORRECT.
21528  110702  077203                  SOB     R2,1$
21529
21530  110704  020027  110742          CMP     R0,#FFBBF1+10   ;IS R0 CORRECT?
21531  110710  001014                  BNE     FFB10           ;BRANCH IF INCORRECT.
21532  110712  022705  000204          CMP     #204,R5         ;IS THE FPS CORRECT?
```

```
21533  110716  001011                  BNE     FFB10           ;BRANCH IF INCORRECT.
21534  110720  000411                  BR      FFBDONE
21535
21536                                  ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
21537  110722  000177                  FFBTP1: 177
21538  110724  167574                          167574
21539  110726  137271                          137271
21540  110730  107675                          107675
21541  110732  177777                  FFBBF1: -1
21542  110734  177777                          -1
21543  110736  177777                          -1
21544  110740  177777                          -1
21545  110742                          FFB10:
  (2)  110742  104000                          EMT                     ;
21546  110744                          FFBDONE:
  (1)  110744  004767  013602                  JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                          ;SEE IF THE USER HAS EXPRESSED
  (1)                                          ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                          ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                          ;THE USER TYPED CONTROL G?).
21547                                  ;*************************************************************************
  (2)                                  ;TEST 534       NEGF, ABSF AND TSTF SOURCE MODE 4 TEST
  (3)                                  ;*************************************************************************
  (2)  110750                          TS534:
```

```
21549
21550  110750  012700  111050                MOV     #GGBTP1,R0              ;SET UP THE DATA BUFFER.
21551  110754  012701  111060                MOV     #GGBBF0,R1
21552  110760  012702  000004                MOV     #4,R2
21553  110764  012021                1$:     MOV     (R0)+,(R1)+
21554  110766  077202                        SOB     R2,1$
21555  110770  012700  000200                MOV     #200,R0         ;SET FD.
21556  110774  170100                        LDFPS   R0
21557  110776  012700  111070                MOV     #GGBBF1,R0      ;SET UP THE OPERAND ADDRESS.
21558  111002  012737  111100  000004        MOV     #GGB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.
21559
21560  111010  170640                GGB2:   ABSD    -(R0)           ;TEST INSTRUCTION.
21561
21562  111012  170205                        STFPS   R5              ;GET FPS.
21563  111014  012701  111060                MOV     #GGBBF0,R1              ;CHECK RESULT.
21564  111020  012702  000004                MOV     #4,R2
21565  111024  005721                1$:     TST     (R1)+
21566  111026  001024                        BNE     GGB10           ;BRANCH IF INCORRECT.
21567  111030  077203                        SOB     R2,1$
21568
21569  111032  020027  111060                CMP     R0,#GGBBF0      ;IS R0 CORRECT?
21570  111036  001020                        BNE     GGB10           ;BRANCH IF INCORRECT.
21571  111040  022705  000204                CMP     #204,R5         ;IS THE FPS CORRECT?
21572  111044  001015                        BNE     GGB10           ;BRANCH IF INCORRECT.
21573  111046  000415                        BR      GGBDONE
21574
21575                                ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
21576  111050  000177                GGBTP1: 177
21577  111052  117273                        117273
21578  111054  147576                        147576
21579  111056  177071                        177071
21580  111060  177777                GGBBF0: -1
21581  111062  177777                        -1
21582  111064  177777                        -1
21583  111066  177777                        -1
21584  111070  177777                GGBBF1: -1
21585  111072  177777                        -1
21586  111074  177777                        -1
21587  111076  177777                        -1
21588  111100                        GGB10:
  (2)  111100  104000                        EMT                     ;
21589  111102                        GGBDONE:
  (1)  111102  004767  013444                JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                               ;SEE IF THE USER HAS EXPRESSED
  (1)                                                               ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                               ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                               ;THE USER TYPED CONTROL G?).
21590                           ;*****************************************************************************
  (2)                           ;TEST 535       NEGF, ABSF AND TSTF SOURCE MODE 3 TEST
  (3)                           ;*****************************************************************************
  (2)  111106                   TS535:
21591
21592  111106  012700  111206                MOV     #HHBTP1,R0              ;SET UP THE DATA BUFFER.
21593  111112  012701  111226                MOV     #HHBBF0,R1
21594  111116  012702  000010                MOV     #10,R2
21595  111122  012021                1$:     MOV     (R0)+,(R1)+
```

```
21596  111124  077202                          SOB     R2,1$
21597  111126  012700  000200                  MOV     #200,R0         ;SET FD.
21598  111132  170100                          LDFPS   R0
21599  111134  012700  111236                  MOV     #MMBBF1,R0      ;SET UP THE OPERAND ADDRESS.
21600  111140  012737  111246  000004          MOV     #MMB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.
21601
21602  111146  170630                  MMB2:   ABSD    @(R0)+          ;TEST INSTRUCTION.
21603
21604  111150  170205                          STFPS   R5              ;GET FPS.
21605  111152  012701  111226                  MOV     #MMBBF0,R1              ;CHECK RESULT.
21606  111156  012702  000004                  MOV     #4,R2
21607  111162  005721                  1$:     TST     (R1)+
21608  111164  001030                          BNE     MMB10           ;BRANCH IF INCORRECT.
21609  111166  077203                          SOB     R2,1$
21610  111170  020027  111240                  CMP     R0,#MMBBF1+2    ;IS R0 CORRECT?
21611  111174  001024                          BNE     MMB10           ;BRANCH IF INCORRECT.
21612  111176  022705  000204                  CMP     #204,R5         ;IS THE FPS CORRECT?
21613  111202  001021                          BNE     MMB10           ;BRANCH IF INCORRECT.
21614  111204  000421                          BR      MMBDONE
21615
21616                                  ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
21617  111206  000177                  MMBTP1: 177
21618  111210  147576                          147576
21619  111212  177071                          177071
21620  111214  107576  111226  177777          107576,MMBBF0,-1,-1,-1
       111222  177777  177777
21621  111226  177777                  MMBBF0: -1
21622  111230  177777                          -1
21623  111232  177777                          -1
21624  111234  177777                          -1
21625  111236  177777                  MMBBF1: -1
21626  111240  177777                          -1
21627  111242  177777                          -1
21628  111244  177777                          -1
21629  111246                          MMB10:
  (2)  111246  104000                          EMT                     ;
21630  111250                          MMBDONE:
  (1)  111250  004767  013276                  JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
21631                           ;*****************************************************************************
  (2)                           ;TEST 536       NEGF, ABSF AND TSTF SOURCE MODE 5 TEST
  (3)                           ;*****************************************************************************
  (2)  111254                   TS536:
21632
21633  111254  012700  111354                  MOV     #IIBTP1,R0              ;SET UP THE DATA BUFFER.
21634  111260  012701  111374                  MOV     #IIBBF0,R1
21635  111264  012702  000010                  MOV     #10,R2
21636  111270  012021                  1$:     MOV     (R0)+,(R1)+
21637  111272  077202                          SOB     R2,1$
21638  111274  012700  000200                  MOV     #200,R0         ;SET FD.
21639  111300  170100                          LDFPS   R0
21640  111302  012700  111406                  MOV     #IIBBF1+2,R0    ;SET UP THE OPERAND ADDRESS.
21641  111306  012737  111414  000004          MOV     #IIB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.
```

```
21642
21643  111314  170750                  IIB2:   NEGD    @-(R0)          ;TEST INSTRUCTION.
21644
21645  111316  170205                          STFPS   R5              ;GET FPS.
21646  111320  012701  111374                  MOV     #IIBBF0,R1              ;CHECK RESULT.
21647  111324  012702  000004                  MOV     #4,R2
21648  111330  005721                  1$:     TST     (R1)+
21649  111332  001030                          BNE     IIB10           ;BRANCH IF INCORRECT.
21650  111334  077203                          SOB     R2,1$
21651  111336  020027  111404                  CMP     R0,#IIBBF1      ;IS R0 CORRECT?
21652  111342  001024                          BNE     IIB10           ;BRANCH IF INCORRECT.
21653  111344  022705  000204                  CMP     #204,R5         ;IS THE FPS CORRECT?
21654  111350  001021                          BNE     IIB10           ;BRANCH IF INCORRECT.
21655  111352  000421                          BR      IIBDONE
21656
21657                                  ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
21658  111354  000176                  IIBTP1: 176
21659  111356  177074                          177074
21660  111360  127374                          127374
21661  111362  157677  111374  177777          157677,IIBBF0,-1,-1,-1
       111370  177777  177777
21662  111374  177777                  IIBBF0: -1
21663  111376  177777                          -1
21664  111400  177777                          -1
21665  111402  177777                          -1
21666  111404  177777                  IIBBF1: -1
21667  111406  177777                          -1
21668  111410  177777                          -1
21669  111412  177777                          -1
21670
21671  111414                          IIB10:
  (2)  111414  104000                          EMT                     ;
21672  111416                          IIBDONE:
  (1)  111416  004767  013130                  JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
21673                                   ;*****************************************************************************
  (2)                                   ;TEST 537       NEGF, ABSF AND TSTF SOURCE MODE 6 TEST
  (3)                                   ;*****************************************************************************
  (2)  111422                           TS537:
21674
21675  111422  012700  111524                  MOV     #JJBTP1,R0              ;SET UP THE DATA BUFFER.
21676  111426  012701  111536                  MOV     #JJBBF0,R1
21677  111432  012702  000004                  MOV     #4,R2
21678  111436  012021                  1$:     MOV     (R0)+,(R1)+
21679  111440  077202                          SOB     R2,1$
21680  111442  012700  000200                  MOV     #200,R0         ;SET FD.
21681  111446  170100                          LDFPS   R0
21682  111450  012700  111527                  MOV     #JJBBF0-7,R0    ;SET UP THE OPERAND ADDRESS.
21683  111454  012737  111546  000004          MOV     #JJB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.
21684
21685  111462  170660  000007          JJB2:   ABSD    7(R0)           ;TEST INSTRUCTION.
21686
21687  111466  170205                          STFPS   R5              ;GET FPS.
```

```
21688  111470  012701  111536              MOV     #JJBBF0,R1              ;CHECK RESULT.
21689  111474  012702  000004              MOV     #4,R2
21690  111500  005721            1$:       TST     (R1)+
21691  111502  001021                      BNE     JJB10           ;BRANCH IF INCORRECT.
21692  111504  077203                      SOB     R2,1$
21693  111506  020027  11152'              CMP     R0,#JJBBF0-7    ;IS R0 CORRECT?
21694  111512  001015                      BNE     JJB10           ;BRANCH IF INCORRECT.
21695  111514  022705  000204              CMP     #204,R5         ;IS THE FPS CORRECT?
21696  111520  001012                      BNE     JJB10           ;BRANCH IF INCORRECT.
21697  111522  000412                      BR      JJBDONE
21698
21699                          ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
21700  111524  000177          JJBTP1: 177
21701  111526  161524                  161524
21702  111530  131273                  131273
21703  111532  107174  000000            107174,
21704  111536  177777          JJBBF0: -1
21705  111540  177777                  -1
21706  111542  177777                  -1
21707  111544  177777                  -1
21708  111546                  JJB10:
  (2)  111546  104000                  EMT                             ;
21709  111550                  JJBDONE:
  (1)  111550  004767  012776          JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                          ;SEE IF THE USER HAS EXPRESSED
  (1)                                                          ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                          ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                          ;THE USER TYPED CONTROL G?).
21710                   ;*****************************************************************************
  (2)                   ;TEST 540       NEGF, ABSF AND TSTF SOURCE MODE 7 TEST
  (3)                   ;*****************************************************************************
  (2)  111554           TS540:
21711
21712  111554  012700  111656              MOV     #KKBTP1,R0              ;SET UP THE DATA BUFFER.
21713  111560  012701  111676              MOV     #KKBBF0,R1
21714  111564  012702  000010              MOV     #10,R2
21715  111570  012021            1$:       MOV     (R0)+,(R1)+
21716  111572  077202                      SOB     R2,1$
21717  111574  012700  000200              MOV     #200,R0         ;SET FD.
21718  111600  170100                      LDFPS   R0
21719  111602  012700  111677              MOV     #KKBBF1-7,R0    ;SET UP THE OPERAND ADDRESS.
21720  111606  012737  111716  000004      MOV     #KKB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.
21721
21722  111614  170770  000007    KKB2:     NEGD    @7(R0)          ;TEST INSTRUCTION.
21723
21724  111620  170205                      STFPS   R5              ;GET FPS.
21725  111622  012701  111676              MOV     #KKBBF0,R1              ;CHECK RESULT.
21726  111626  012702  000004              MOV     #4,R2
21727  111632  005721            1$:       TST     (R1)+
21728  111634  001030                      BNE     KKB10           ;BRANCH IF INCORRECT.
21729  111636  077203                      SOB     R2,1$
21730  111640  020027  111677              CMP     R0,#KKBBF1-7    ;IS R0 CORRECT?
21731  111644  001024                      BNE     KKB10           ;BRANCH IF INCORRECT.
21732  111646  022705  000204              CMP     #204,R5         ;IS THE FPS CORRECT?
21733  111652  001021                      BNE     KKB10           ;BRANCH IF INCORRECT.
21734  111654  000421                      BR      KKBDONE
```

```
21735
21736                                   ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
21737  111656  000177                  KKBTP1: 177
21738  111660  167574                          167574
21739  111662  137271                          137271
21740  111664  107675  111676  177777          107675,KKBBF0,-1,-1,-1
       111672  177777  177777
21741  111676  177777                  KKBBF0: -1
21742  111700  177777                          -1
21743  111702  177777                          -1
21744  111704  177777                          -1
21745  111706  177777                  KKBBF1: -1
21746  111710  177777                          -1
21747  111712  177777                          -1
21748  111714  177777                          -1
21749  111716                          KKB10:
  (2)  111716  104000                          EMT                     ;
21750  111720                          KKBDONE:
  (1)  111720  004767  012626                  JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
21751                                   ;*********************************************************************
  (2)                                   ;TEST 541       NEGF, ABSF AND TSTF SOURCE MODE 6, GR7, TEST
  (3)                                   ;*********************************************************************
  (2)  111724                           TS541:
21752  111724  012700  112014                  MOV     #LLBTP1,R0              ;SET UP THE DATA BUFFER.
21753  111730  012701  112024                  MOV     #LLBBF0,R1
21754  111734  012702  000004                  MOV     #4,R2
21755  111740  012021                  1$:     MOV     (R0)+,(R1)+
21756  111742  077202                          SOB     R2,1$
21757  111744  012700  000200                  MOV     #200,R0         ;SET FD.
21758  111750  170100                          LDFPS   R0
21759  111752  012737  112034  000004          MOV     #LLB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.
21760
21761  111760  170767  000040          LLB2:   NEGD    LLBBF0          ;TEST INSTRUCTION.
21762
21763  111764  170205                          STFPS   R5              ;GET FPS.
21764  111766  012701  112024                  MOV     #LLBBF0,R1              ;CHECK RESULT.
21765  111772  012702  000004                  MOV     #4,R2
21766  111776  005721                  1$:     TST     (R1)+
21767  112000  001015                          BNE     LLB10           ;BRANCH IF INCORRECT.
21768  112002  077203                          SOB     R2,1$
21769  112004  022705  000204                  CMP     #204,R5         ;IS THE FPS CORRECT?
21770  112010  001011                          BNE     LLB10           ;BRANCH IF INCORRECT.
21771  112012  000411                          BR      LLBDONE
21772
21773                                   ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
21774  112014  000127                  LLBTP1: 127
21775  112016  137475                          137475
21776  112020  147372                          147372
21777  112022  117057                          117057
21778  112024  177777                  LLBBF0: -1
21779  112026  177777                          -1
21780  112030  177777                          -1
```

```
21781  112032  177777                          -1
21782
21783  112034                          LLB10:
  (2)  112034  104000                          EMT                     ;
21784  112036                          LLBDONE:
  (1)  112036  004767  012510                  JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
21785                             ;*************************************************************************
  (2)                             ;TEST 542       NEGF, ABSF AND TSTF SOURCE MODE 7, GR7, TEST
  (3)                             ;*************************************************************************
  (2)  112042                     TS542:
21786
21787  112042  012700  112132                  MOV     #MMBTP1,R0              ;SET UP THE DATA BUFFER.
21788  112046  012701  112152                  MOV     #MMBBF0,R1
21789  112052  012702  000010                  MOV     #10,R2
21790  112056  012021                  1$:     MOV     (R0)+,(R1)+
21791  112060  077202                          SOB     R2,1$
21792  112062  012700  000200                  MOV     #200,R0         ;SET FD.
21793  112066  170100                          LDFPS   R0
21794  112070  012737  112172  000004          MOV     #MMB10,@#ERRVECT ;SET UP VECTOR 4 IN CASE OF AN ERROR.
21795
21796  112076  170677  000060          MMB2:   ABSD    @#MMBBF1        ;TEST INSTRUCTION.
21797
21798  112102  170205                          STFPS   R5              ;GET FPS.
21799  112104  012701  112152                  MOV     #MMBBF0,R1              ;CHECK RESULT.
21800  112110  012702  000004                  MOV     #4,R2
21801  112114  005721                  1$:     TST     (R1)+
21802  112116  001025                          BNE     MMB10           ;BRANCH IF INCORRECT.
21803  112120  077203                          SOB     R2,1$
21804  112122  022705  000204                  CMP     #204,R5         ;IS THE FPS CORRECT?
21805  112126  001021                          BNE     MMB10           ;BRANCH IF INCORRECT.
21806  112130  000421                          BR      MMBDONE
21807
21808                             ;THESE ARE TEST DATA TABLES AND DATA BUFFER.
21809  112132  000137              MMBTP1: 137
21810  112134  045607                      045607
21811  112136  101230                      101230
21812  112140  045607  112152  177777      45607,MMBBF0,-1,-1,-1
       112146  177777  177777
21813  112152  177777              MMBBF0: -1
21814  112154  177777                      -1
21815  112156  177777                      -1
21816  112160  177777                      -1
21817  112162  177777              MMBBF1: -1
21818  112164  177777                      -1
21819  112166  177777                      -1
21820  112170  177777                      -1
21821
21822  112172                          MMB10:
  (2)  112172  104000                          EMT                     ;
21823  112174                          MMBDONE:
  (1)  112174  004767  012352                  JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
```

```
  (1)                                                                   ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                   ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                   ;THE USER TYPED CONTROL G?).
21830                                   ;*******************************************************************************
  (2)                                   ;TEST 543       SPECIAL DEST, MODE 0, TEST
  (3)                                   ;*******************************************************************************
  (2)  112200                           TS543:
21831
21832  112200  012700  000200                   MOV     #200,R0         ;SET FD.
21833  112204  170100                           LDFPS   R0
21834  112206  012700  112266                   MOV     #NNBTP1,R0              ;SET UP AC0.
21835  112212  172410                           LDD     (R0),AC0
21836  112214  170700                   NNB2:   NEGD    AC0             ;TEST INSTRUCTION.
21837
21838  112216  170205                           STFPS   R5              ;GET FPS.
21839  112220  012700  000200                   MOV     #200,R0         ;SET FD.
21840  112224  170100                           LDFPS   R0
21841  112226  012700  112306                   MOV     #NNBBF0,R0              ;GET THE RESULT.
21842  112232  174010                           STD     AC0,(R0)
21843  112234  012700  112306                   MOV     #NNBBF0,R0              ;IS THE RESULT CORRECT?
21844  112240  012701  112276                   MOV     #NNBTP2,R1
21845  112244  012702  000004                   MOV     #4,R2
21846  112250  022021                   1$:     CMP     (R0)+,(R1)+
21847  112252  001021                           BNE     NNB10           ;BRANCH IF INCORRECT.
21848  112254  077203                           SOB     R2,1$
21849  112256  022705  000210                   CMP     #210,R5         ;IS THE FPS CORRECT?
21850  112262  001015                           BNE     NNB10           ;BRANCH IF INCORRECT.
21851  112264  000415                           BR      NNBDONE
21852
21853                                   ;THESE ARE DATA TABLES AND A DATA BUFFER.
21854  112266  013572                   NNBTP1: 013572
21855  112270  046013                           46013
21856  112272  057246                           57246
21857  112274  013570                           013570
21858  112276  113572                   NNBTP2: 113572
21859  112300  046013                           46013
21860  112302  057246                           57246
21861  112304  013570                           013570
21862  112306  000000                   NNBBF0: 0
21863  112310  000000                           0
21864  112312  000000                           0
21865  112314  000000                           0
21866
21867  112316                           NNB10:
  (2)  112316  104000                           EMT                     ;
21868  112320                           NNBDONE:
  (1)  112320  004767  012226                   JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                   ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                   ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                   ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                   ;THE USER TYPED CONTROL G?).
21869                                   ;*******************************************************************************
  (2)                                   ;TEST 544       SPECIAL DEST, MODE 1, TEST
  (3)                                   ;*******************************************************************************
  (2)  112324                           TS544:
21870
```

```
21871  112324  012701  112426                  MOV     #OOBTP1,R1      ;SET UP THE DATA BUFFER.
21872  112330  012700  112436                  MOV     #OOBTP2,R0
21873  112334  012702  000004                  MOV     #4,R2
21874  112340  012021                  1$:     MOV     (R0)+,(R1)+
21875  112342  077202                          SOB     R2,1$
21876  112344  012700  112426                  MOV     #OOBTP1,R0
21877  112350  042710  100000                  BIC     #100000,(R0)    ;MAKE OPERAND POSITIVE.
21878  112354  012701  000200                  MOV     #200,R1         ;SET FD.
21879  112360  170101                          LDFPS   R1
21880
21881  112362  170710                  OOB2:   NEGD    (R0)            ;TEST INSTRUCTION.
21882  112364  170205                          STFPS   R5              ;GET FPS.
21883  112366  012701  112426                  MOV     #OOBTP1,R1              ;IS THE RESULT CORRECT.
21884  112372  012702  112436                  MOV     #OOBTP2,R2
21885  112376  012703  000004                  MOV     #4,R3
21886  112402  022122                  1$:     CMP     (R1)+,(R2)+
21887  112404  001020                          BNE     OOB10           ;BRANCH IF INCORRECT.
21888  112406  077303                          SOB     R3,1$
21889  112410  022700  112426                  CMP     #OOBTP1,R0      ;IS R0 CORRECT.
21890  112414  001014                          BNE     OOB10           ;BRANCH IF INCORRECT.
21891  112416  022705  000210                  CMP     #210,R5         ;IS THE FPS CORRECT?
21892  112422  001011                          BNE     OOB10           ;BRANCH IF INCORRECT.
21893  112424  000411                          BR      OOBDONE
21894
21895                                  ;THESE ARE DATA TABLES AND A DATA BUFFER.
21896  112426  023245                  OOBTP1: 023245
21897  112430  026720                          26720
21898  112432  122324                          122324
21899  112434  052672                          52672
21900  112436  123245                  OOBTP2: 123245
21901  112440  026720                          26720
21902  112442  122324                          122324
21903  112444  052672                          52672
21904
21905  112446                          OOB10:
  (2)  112446  104000                          EMT                     ;
21906  112450                          OOBDONE:
  (1)  112450  004767  012076                  JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
21907                           ;*************************************************************************
  (2)                           ;TEST 545       SPECIAL DEST, MODE 2, TEST
  (3)                           ;*************************************************************************
  (2)  112454                   TS545:
21908
21909  112454  012701  112556                  MOV     #PPBTP1,R1      ;SET UP THE DATA BUFFER.
21910  112460  012700  112566                  MOV     #PPBTP2,R0
21911  112464  012702  000004                  MOV     #4,R2
21912  112470  012021                  1$:     MOV     (R0)+,(R1)+
21913  112472  077202                          SOB     R2,1$
21914  112474  012700  112556                  MOV     #PPBTP1,R0
21915  112500  042710  100000                  BIC     #100000,(R0)    ;MAKE OPERAND POSITIVE.
21916  112504  012701  000200                  MOV     #200,R1         ;SET FD.
21917  112510  170101                          LDFPS   R1
```

```
21918
21919  112512  170720                  PPB2:   NEGD    (R0)+           ;TEST INSTRUCTION.
21920
21921  112514  170205                          STFPS   R5              ;GET FPS.
21922  112516  012701  112556                  MOV     #PPBTP1,R1              ;IS THE RESULT CORRECT.
21923  112522  012702  112566                  MOV     #PPBTP2,R2
21924  112526  012703  000004                  MOV     #4,R3
21925  112532  022122                  1$:     CMP     (R1)+,(R2)+
21926  112534  001020                          BNE     PPB10           ;BRANCH IF INCORRECT.
21927  112536  077303                          SOB     R3,1$
21928  112540  022700  112566                  CMP     #PPBTP1+10,R0   ;IS R0 CORRECT.
21929  112544  001014                          BNE     PPB10           ;BRANCH IF INCORRECT.
21930  112546  022705  000210                  CMP     #210,R5         ;IS THE FPS CORRECT?
21931  112552  001011                          BNE     PPB10           ;BRANCH IF INCORRECT.
21932  112554  000411                          BR      PPBDONE
21933
21934                                  ;THESE ARE DATA TABLES AND A DATA BUFFER.
21935  112556  023245                  PPBTP1: 023245
21936  112560  026720                          26720
21937  112562  122324                          122324
21938  112564  052672                          52672
21939  112566  123245                  PPBTP2: 123245
21940  112570  026720                          26720
21941  112572  122324                          122324
21942  112574  052672                          52672
21943
21944  112576                          PPB10:
  (2)  112576  104000                          EMT                     ;
21945  112600                          PPBDONE:
  (1)  112600  004767  011746                  JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
21946                             ;*************************************************************************
  (2)                             ;TEST 546       SPECIAL DEST, MODE 4, TEST
  (3)                             ;*************************************************************************
  (2)  112604                     TS546:
21947  112604  012701  112710                  MOV     #QQBTP1,R1      ;SET UP THE DATA BUFFER.
21948  112610  012700  112730                  MOV     #QQBTP2,R0
21949  112614  012702  000004                  MOV     #4,R2
21950  112620  012021                  1$:     MOV     (R0)+,(R1)+
21951  112622  077202                          SOB     R2,1$
21952  112624  012700  112720                  MOV     #QQBTP1+10,R0
21953  112630  042760  100000  177770          BIC     #100000,-10(R0) ;MAKE OPERAND POSITIVE.
21954  112636  012701  000200                  MOV     #200,R1         ;SET FD.
21955  112642  170101                          LDFPS   R1
21956
21957  112644  170740                  QQB2:   NEGD    -(R0)           ;TEST INSTRUCTION.
21958
21959  112646  170205                          STFPS   R5              ;GET FPS.
21960  112650  012701  112710                  MOV     #QQBTP1,R1              ;IS THE RESULT CORRECT.
21961  112654  012702  112730                  MOV     #QQBTP2,R2
21962  112660  012703  000004                  MOV     #4,R3
21963  112664  022122                  1$:     CMP     (R1)+,(R2)+
21964  112666  001024                          BNE     QQB10           ;BRANCH IF INCORRECT.
```

```
21965  112670  077303                          SOB     R3,1$
21966  112672  022700  112710                  CMP     #QQBTP1,R0      ;IS R0 CORRECT.
21967  112676  001020                          BNE     QQB10           ;BRANCH IF INCORRECT.
21968  112700  022705  000210                  CMP     #210,R5         ;IS THE FPS CORRECT?
21969  112704  001015                          BNE     QQB10           ;BRANCH IF INCORRECT.
21970  112706  000415                          BR      QQBDONE
21971
21972                                  ;THESE ARE DATA TABLES AND A DATA BUFFER.
21973  112710  023245                  QQBTP1: 023245
21974  112712  026720                          26720
21975  112714  122324                          122324
21976  112716  052672                          52672
21977  112720  177777  177777  177777          .WORD   -1,-1,-1,-1
       112726  177777
21978  112730  123245                  QQBTP2: 123245
21979  112732  026720                          26720
21980  112734  122324                          122324
21981  112736  052672                          52672
21982
21983  112740                          QQB10:
  (2)  112740  104000                          EMT                     ;
21984  112742                          QQBDONE:
  (1)  112742  004767  011604                  JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
21985
21986                           ;*****************************************************************************
  (2)                           ;TEST 547       SPECIAL DEST, MODE 3, TEST
  (3)                           ;*****************************************************************************
  (2)  112746                   TS547:
21987
21988  112746  012701  113056                  MOV     #RRBTP1,R1      ;SET UP THE DATA BUFFER.
21989  112752  012700  113066                  MOV     #RRBTP2,R0
21990  112756  012702  000004                  MOV     #4,R2
21991  112762  012021                  1$:     MOV     (R0)+,(R1)+
21992  112764  077202                          SOB     R2,1$
21993  112766  012700  113076                  MOV     #RRBTP3,R0
21994  112772  012710  113056                  MOV     #RRBTP1,(R0)
21995  112776  042737  100000  113056          BIC     #100000,@#RRBTP1        ;MAKE THE OPERAND POSITIVE.
21996  113004  012701  000200                  MOV     #200,R1         ;SET FD.
21997  113010  170101                          LDFPS   R1
21998
21999  113012  170730                  RRB2:   NEGD    @(R0)+          ;TEST INSTRUCTION.
22000
22001  113014  170205                          STFPS   R5              ;GET FPS.
22002  113016  012701  113056                  MOV     #RRBTP1,R1              ;IS THE RESULT CORRECT.
22003  113022  012702  '13066                 MOV     #RRBTP2,R2
22004  113026  012703  ↓00004                 MOV     #4,R3
22005  113032  022122                  1$:     CMP     (R1)+,(R2)+
22006  113034  001021                          BNE     RRB10           ;BRANCH IF INCORRECT.
22007  113036  077303                          SOB     R3,1$
22008  113040  022700  113100                  CMP     #RRBTP3+2,R0    ;IS R0 CORRECT.
22009  113044  001015                          BNE     RRB10           ;BRANCH IF INCORRECT.
22010  113046  022705  000210                  CMP     #210,R5         ;IS THE FPS CORRECT?
```

```
22011  113052  001012                        BNE     RRB10           ;BRANCH IF INCORRECT.
22012  113054  000412                        BR      RRBDONE
22013
22014                                  ;THESE ARE DATA TABLES AND A DATA BUFFER.
22015  113056  023245                  RRBTP1: 023245
22016  113060  026720                          26720
22017  113062  122324                          122324
22018  113064  052672                          52672
22019  113066  123245                  RRBTP2: 123245
22020  113070  026720                          26720
22021  113072  123324                          123324
22022  113074  052672                          52672
22023  113076  113056                  RRBTP3: RRBTP1
22024
22025  113100                          RRB10:
  (2)  113100  104000                          EMT                     ;
22026  113102                          RRBDONE:
  (1)  113102  004767  011444                  JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
22027
22028                                  ;*************************************************************************
  (2)                                  ;TEST 550       SPECIAL DEST, MODE 5, TEST
  (3)                                  ;*************************************************************************
  (2)  113106                          TS550:
22029  113106  012701  113220                  MOV     #SSBTP1,R1      ;SET UP THE DATA BUFFER.
22030  113112  012700  113230                  MOV     #SSBTP2,R0
22031  113116  012702  000004                  MOV     #4,R2
22032  113122  012021                  1$:     MOV     (R0)+,(R1)+
22033  113124  077202                          SOB     R2,1$
22034  113126  012700  113242                  MOV     #SSBTP3+2,R0
22035  113132  012760  113220  177776          MOV     #SSBTP1,-2(R0)
22036  113140  042737  100000  113220          BIC     #100000,@#SSBTP1        ;MAKE THE OPERAND POSITIVE.
22037  113146  012701  000200                  MOV     #200,R1         ;SET FD.
22038  113152  170101                          LDFPS   R1
22039
22040  113154  170750                  SSB2:   NEGD    @-(R0)          ;TEST INSTRUCTION.
22041
22042  113156  170205                          STFPS   R5              ;GET FPS.
22043  113160  012701  113220                  MOV     #SSBTP1,R1              ;IS THE RESULT CORRECT.
22044  113164  012702  113230                  MOV     #SSBTP2,R2
22045  113170  012703  000004                  MOV     #4,R3
22046  113174  022122                  1$:     CMP     (R1)+,(R2)+
22047  113176  001021                          BNE     SSB10           ;BRANCH IF INCORRECT.
22048  113200  077303                          SOB     R3,1$
22049  113202  022700  113240                  CMP     #SSBTP3,R0      ;IS R0 CORRECT.
22050  113206  001015                          BNE     SSB10           ;BRANCH IF INCORRECT.
22051  113210  022705  000210                  CMP     #210,R5         ;IS THE FPS CORRECT?
22052  113214  001012                          BNE     SSB10           ;BRANCH IF INCORRECT.
22053  113216  000412                          BR      SSBDONE
22054
22055                                  ;THESE ARE DATA TABLES AND A DATA BUFFER.
22056  113220  023245                  SSBTP1: 023245
22057  113222  026720                          26720
```

```
22058  113224  122324                  122324
22059  113226  052672                  52672
22060  113230  123245          SSBTP2: 123245
22061  113232  026270                  26270
22062  113234  122324                  122324
22063  113236  052672                  52672
22064  113240  113220          SSBTP3: SSBTP1
22065
22066  113242                  SSB10:
  (2)  113242  104000                  EMT                     ;
22067  113244                  SSBDONE:
  (1)  113244  004767  011302          JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                          ;SEE IF THE USER HAS EXPRESSED
  (1)                                                          ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                          ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                          ;THE USER TYPED CONTROL G?).
22068                           ;*************************************************************************
  (2)                           ;TEST 551       SPECIAL DEST, FLOATING MODE 2, TEST
  (3)                           ;*************************************************************************
  (2)  113250                   TS551:
22069  113250  012701  113352          MOV     #TTBTP1,R1      ;SET UP THE DATA BUFFER.
22070  113254  012700  113362          MOV     #TTBTP2,R0
22071  113260  012702  000004          MOV     #4,R2
22072  113264  012021          1$:     MOV     (R0)+,(R1)+
22073  113266  077202                  SOB     R2,1$
22074  113270  012700  113352          MOV     #TTBTP1,R0
22075  113274  042710  100000          BIC     #100000,(R0)    ;MAKE OPERAND POSITIVE.
22076  113300  012701  000000          MOV     #000,R1         ;SET FD.
22077  113304  170101                  LDFPS   R1
22078
22079  113306  170720          TTB2:   NEGF    (R0)+           ;TEST INSTRUCTION.
22080
22081  113310  170205                  STFPS   R5              ;GET FPS.
22082  113312  012701  113352          MOV     #TTBTP1,R1              ;IS THE RESULT CORRECT.
22083  113316  012702  113362          MOV     #TTBTP2,R2
22084  113322  012703  000004          MOV     #4,R3
22085  113326  022122          1$:     CMP     (R1)+,(R2)+
22086  113330  001020                  BNE     TTB10           ;BRANCH IF INCORRECT.
22087  113332  077303                  SOB     R3,1$
22088  113334  022700  113356          CMP     #TTBTP1+4,R0    ;IS R0 CORRECT.
22089  113340  001014                  BNE     TTB10           ;BRANCH IF INCORRECT.
22090  113342  022705  000010          CMP     #010,R5         ;IS THE FPS CORRECT?
22091  113346  001011                  BNE     TTB10           ;BRANCH IF INCORRECT.
22092  113350  000411                  BR      TTBDONE
22093
22094                           ;THESE ARE DATA TABLES AND A DATA BUFFER.
22095  113352  023245          TTBTP1: 023245
22096  113354  026720                  26720
22097  113356  122324                  122324
22098  113360  052672                  52672
22099  113362  123245          TTBTP2: 123245
22100  113364  026720                  26720
22101  113366  122324                  122324
22102  113370  052672                  52672
22103
22104  113372                  TTB10:
```

```
  (2)  113372  104000                    EMT                         ;
22105  113374                          TTBDONE:
  (1)  113374  004767  011152                JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                 ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                 ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                 ;THE USER TYPED CONTROL G?).
22106                             ;*************************************************************************
  (2)                             ;TEST 552       SPECIAL DEST, MODE2, GR7 (IMMEDIATE), TEST
  (3)                             ;*************************************************************************
  (2)  113400                     TS552:
22107  113400  012700  113516              MOV     #UUBTP2,R0
22108  113404  012701  113444              MOV     #UUBTP1,R1      ;SET UP THE DATA BUFFER.
22109  113410  012702  000004              MOV     #4,R2
22110  113414  012021                1$:    MOV     (R0)+,(R1)+
22111  113416  077202                       SOB     R2,1$
22112  113420  012700  113444              MOV     #UUBTP1,R0
22113  113424  042737  100000  113444      BIC     #100000,@#UUBTP1        ;MAKE THE OPERAND POSITIVE.
22114  113432  012701  000200              MOV     #200,R1         ;SET FD.
22115  113436  170101                       LDFPS   R1
22116  113440  005001                       CLR     R1
22117
22118  113442  170727                UUB2:  NEGD    (R7)+           ;TEST INSTRUCTION.
22119  113444  005201  005201  005201 UUBTP1: 5201,5201,5201,5201
       113452  005201
22120                             ;NOTE THAT AFTER EXECUTING THIS INSTRUCTION R1 SHOULD CONTAIN 3.
22121  113454  170205                       STFPS   R5              ;GET FPS.
22122  113456  012703  113444              MOV     #UUBTP1,R3      ;IS THE RESULT CORRECT.
22123  113462  012702  113516              MOV     #UUBTP2,R2
22124  113466  012704  000004              MOV     #4,R4
22125  113472  022322                1$:    CMP     (R3)+,(R2)+
22126  113474  001014                       BNE     UUB10           ;BRANCH IF INCORRECT.
22127  113476  077403                       SOB     R4,1$
22128  113500  022701  000003              CMP     #3,R1           ;WAS R1 INCREMENTED CORRECTLY.
22129  113504  001010                       BNE     UUB10           ;BRANCH IF INCORRECT.
22130  113506  022705  000210              CMP     #210,R5         ;IS THE FPS CORRECT?
22131  113512  001005                       BNE     UUB10           ;BRANCH IF INCORRECT.
22132  113514  000405                       BR      UUBDONE
22133
22134                             ;THESE ARE DATA TABLE.
22135  113516  105201                UUBTP2: 105201
22136  113520  005201                        5201
22137  113522  005201                        5201
22138  113524  005201                        5201
22139
22140  113526                          UUB10:
  (2)  113526  104000                    EMT                         ;
22141  113530                          UUBDONE:
  (1)  113530  004767  011016                JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                 ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                 ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                 ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                 ;THE USER TYPED CONTROL G?).
22142                             ;*************************************************************************
  (2)                             ;TEST 553       SPECIAL DEST, MODE 6, TEST
  (3)                             ;*************************************************************************
```

```
  (2)  113534                              TS553:
22143  113534  012701  113650                      MOV     #XXBTP1,R1      ;SET UP THE DATA BUFFER.
22144  113540  012700  113660                      MOV     #XXBTP2,R0
22145  113544  012702  000004                      MOV     #4,R2
22146  113550  012021                      1$:     MOV     (R0)+,(R1)+
22147  113552  077202                              SOB     R2,1$
22148  113554  012700  106447                      MOV     #XXBTP1-5201,R0
22149  113560  042737  100000  113650              BIC     #100000,@#XXBTP1;MAKE OPERAND POSITIVE.
22150  113566  012701  000200                      MOV     #200,R1         ;SET FD.
22151  113572  170101                              LDFPS   R1
22152
22153  113574  005001                              CLR     R1
22154  113576  170760  005201              XXB2:   NEGD    5201(R0)                ;TEST INSTRUCTION.
22155
22156  113602  170205                              STFPS   R5              ;GET FPS.
22157  113604  005701                              TST     R1
22158  113606  001030                              BNE     XXB10           ;WAS THE PC CORRECT AFTER EXECUTION?
22159  113610  012701  113650                      MOV     #XXBTP1,R1              ;IS THE RESULT CORRECT.
22160  113614  012702  113660                      MOV     #XXBTP2,R2
22161  113620  012703  000004                      MOV     #4,R3
22162  113624  022122                      1$:     CMP     (R1)+,(R2)+
22163  113626  001020                              BNE     XXB10           ;BRANCH IF INCORRECT.
22164  113630  077303                              SOB     R3,1$
22165  113632  022700  106447                      CMP     #XXBTP1-5201,R0 ;IS R0 CORRECT.
22166  113636  001014                              BNE     XXB10           ;BRANCH IF INCORRECT.
22167  113640  022705  000210                      CMP     #210,R5         ;IS THE FPS CORRECT?
22168  113644  001011                              BNE     XXB10           ;BRANCH IF INCORRECT.
22169  113646  000411                              BR      XXBDONE
22170
22171                                   ;THESE ARE DATA TABLES AND A DATA BUFFER.
22172  113650  023245                  XXBTP1: 023245
22173  113652  026720                          26720
22174  113654  122324                          122324
22175  113656  052672                          52672
22176  113660  123245                  XXBTP2: 123245
22177  113662  026720                          26720
22178  113664  122324                          122324
22179  113666  052672                          52672
22180
22181
22182  113670                          XXB10:
  (2)  113670  104000                          EMT                     ;
22183  113672                          XXBDONE:
  (1)  113672  004767  010654                  JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
22184
22185                                   ;*****************************************************************************
  (2)                                   ;TEST 554       SPECIAL DEST, MODE 7, TEST
  (3)                                   ;*****************************************************************************
  (2)  113676                           TS554:
22186
22187  113676  012701  114020                  MOV     #YYBTP1,R1      ;SET UP THE DATA BUFFER.
22188  113702  012700  114030                  MOV     #YYBTP2,R0
```

```
22189  113706  012702  000004              MOV     #4,R2
22190  113712  012021                1$:   MOV     (R0)+,(R1)+
22191  113714  077202                      SOB     R2,1$
22192  113716  012700  106637              MOV     #YYBTP3-5201,R0
22193  113722  012760  114020  005201      MOV     #YYBTP1,5201(R0)
22194  113730  042737  100000  114020      BIC     #100000,@#YYBTP1        ;MAKE THE OPERAND POSITIVE.
22195  113736  012701  000200              MOV     #200,R1         ;SET FD.
22196  113742  170101                      LDFPS   R1
22197
22198  113744  005001                      CLR     R1
22199  113746  170770  005201        YYB2: NEGD    @5201(R0)               ;TEST INSTRUCTION.
22200
22201  113752  170205                      STFPS   R5              ;GET FPS.
22202  113754  005701                      TST     R1              ;WAS THE PC CORRECT AFTER EXECUTION?
22203  113756  001031                      BNE     YYB10
22204  113760  012701  114020              MOV     #YYBTP1,R1              ;IS THE RESULT CORRECT.
22205  113764  012702  114030              MOV     #YYBTP2,R2
22206  113770  012703  000004              MOV     #4,R3
22207  113774  022122                1$:   CMP     (R1)+,(R2)+
22208  113776  001021                      BNE     YYB10           ;BRANCH IF INCORRECT.
22209  114000  077303                      SOB     R3,1$
22210  114002  022700  106637              CMP     #YYBTP3-5201,R0 ;IS R0 CORRECT.
22211  114006  001015                      BNE     YYB10           ;BRANCH IF INCORRECT.
22212  114010  022705  000210              CMP     #210,R5         ;IS THE FPS CORRECT?
22213  114014  001012                      BNE     YYB10           ;BRANCH IF INCORRECT.
22214  114016  000412                      BR      YYBDONE
22215
22216                                ;THESE ARE DATA TABLES AND A DATA BUFFER.
22217  114020  023245                YYBTP1: 023245
22218  114022  026720                        26720
22219  114024  122324                        122324
22220  114026  052672                        52672
22221  114030  123245                YYBTP2: 123245
22222  114032  026720                        26720
22223  114034  123324                        123324
22224  114036  052672                        52672
22225  114040  114020                YYBTP3: YYBTP1
22226  114042                        YYB10:
  (2)  114042  104000                        EMT                     ;
22227
22228  114044                        YYBDONE:
  (1)  114044  004767  010502                JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                ;THE USER TYPED CONTROL G?).
22234                           ;*****************************************************************************
  (2)                           ;TEST 555       NEGD, ABSD AND TSTD TEST
  (3)                           ;*****************************************************************************
  (2)  114050                   TS555:
22235                           ;TEST NEGD WITH POS NONZERO OPERAND
22236  114050  004767  000526  WMB1:  JSR     PC,NATSUB
22237  114054  000000            1$:   0                               ;FLAG=NEGD.
22238  114056  016341            2$:   16341                           ;OPERAND.
22239  114060  055772                  55772
22240  114062  021133                  21133
```

```
114064  055447                          55447
114066  116341                  3$:     116341                  ;RESULT.
114070  055772                          55772
114072  021133                          21133
114074  055447                          55447
114076  016341                  4$:     16341                   ;ERROR RES.
114100  055772                          55772
114102  021133                          21133
114104  055447                          55447
114106  000207                  5$:     207                     ;FPS BEFORE EXECUTION.
114110  000210                          210                     ;FPS AFTER EXECUTION.
114112  000200                          200                     ;ERROR FPS.
114114  177777                          -1                      ;FEC
                                ;TEST NEGD WITH NEG OPERAND.
114116  004767  000460          WWB2:   JSR     PC,NATSUB
114122  000000                  1$:     0                       ;FLAG=NEGD.
114124  152525                  2$:     152525                  ;OPERAND.
114126  053545                          53545
114130  055565                          55565
114132  057505                          57505
114134  052525                  3$:     52525                   ;RESULT.
114136  053545                          53545
114140  055565                          55565
114142  057505                          57505
114144  152525                  4$:     152525                  ;ERROR RES.
114146  053545                          53545
114150  055565                          55565
114152  057505                          57505
114154  000217                  5$:     217                     ;FPS BEFORE EXECUTION.
114156  000200                          200                     ;FPS AFTER EXECUTION.
114160  000210                          210                     ;ERROR FPS.
114162  177777                          -1                      ;FEC
                                ;TEST ABSD WITH POSITIVE OPERAND
114164  004767  000412          WWB3:   JSR     PC,NATSUB
114170  000001                  1$:     1                       ;FLAG=ABSD.
114172  060705                  2$:     60705                   ;OPERAND.
114174  124735                          124735
114176  060124                          60124
114200  073560                          73560
114202  060705                  3$:     60705                   ;RESULT.
114204  124735                          124735
114206  060124                          60124
114210  073560                          73560
114212  160705                  4$:     160705                  ;ERROR RES.
114214  124735                          124735
114216  060124                          60124
114220  073560                          73560
114222  000217                  5$:     217                     ;FPS BEFORE EXECUTION.
114224  000200                          200                     ;FPS AFTER EXECUTION.
114226  000210                          210                     ;ERROR FPS.
114230  177777                          -1                      ;EITHER BUT OP1B
                                ;TEST ABSD WITH NEG. OPERAND
114232  004767  000344          WWB4:   JSR     PC,NATSUB
114236  000001                  1$:     1                       ;FLAG=ABSD.
114240  154345                  2$:     154345                  ;OPERAND.
114242  076567                          76567
```

```
22297  114244  032123                      32123
22298  114246  043234                      43234
22299  114250  054345              3$:     54345                   ;RESULT.
22300  114252  076567                      76567
22301  114254  032123                      32123
22302  114256  043234                      43234
22303  114260  154345              4$:     154345                  ;ERROR RES.
22304  114262  076567                      76567
22305  114264  032123                      32123
22306  114266  043234                      43234
22307  114270  000217              5$:     217                     ;FPS BEFORE EXECUTION.
22308  114272  000200                      200                     ;FPS AFTER EXECUTION.
22309  114274  177777                      -1                      ;ERROR FPS.
22310  114276  177777                      -1
22311                              ;TEST WITH POSITIVE OP
22312  114300  004767  000276      WMB5:   JSR     PC,NATSUB
22313  114304  000002              1$:     2                       ;FLAG=TSTD.
22314  114306  012321              2$:     12321                   ;OPERAND.
22315  114310  045654                      45654
22316  114312  070107                      70107
22317  114314  034543                      34543
22318  114316  012321              3$:     12321                   ;RESULT.
22319  114320  045654                      45654
22320  114322  070107                      70107
22321  114324  034543                      34543
22322  114326  112321              4$:     112321                  ;ERROR RES.
22323  114330  045654                      45654
22324  114332  070107                      70107
22325  114334  034543                      34543
22326  114336  000217              5$:     217                     ;FPS BEFORE EXECUTION.
22327  114340  000200                      200                     ;FPS AFTER EXECUTION.
22328  114342  000210                      210                     ;ERROR FPS.
22329  114344  177777                      -1
22330                              ;TEST TSTD WITH NEG OP
22331  114346  004767  000230      WMB6:   JSR     PC,NATSUB
22332  114352  000002              1$:     2                       ;FLAG=TSTD.
22333  114354  123765              2$:     123765                  ;OPERAND.
22334  114356  023407                      23407
22335  114360  034510                      34510
22336  114362  045621                      45621
22337  114364  123765              3$:     123765                  ;RESULT.
22338  114366  023407                      23407
22339  114370  034510                      34510
22340  114372  045621                      45621
22341  114374  023765              4$:     23765                   ;ERROR RES.
22342  114376  023407                      23407
22343  114400  034510                      34510
22344  114402  045621                      45621
22345  114404  000207              5$:     207                     ;FPS BEFORE EXECUTION.
22346  114406  000210                      210                     ;FPS AFTER EXECUTION.
22347  114410  000200                      200                     ;ERROR FPS.
22348  114412  177777                      -1
22349                              ;TEST TSTD 0 OP
22350  114414  004767  000162      WMB7:   JSR     PC,NATSUB
22351  114420  000002              1$:     2                       ;FLAG=TSTD.
22352  114422  000175              2$:     175                     ;OPERAND.
```

```
22353  114424  176737                          176737
22354  114426  071727                          71727
22355  114430  037574                          37574
22356  114432  000175                  3$:     175                     ;RESULT.
22357  114434  176737                          176737
22358  114436  071727                          71727
22359  114440  037574                          37574
22360  114442  000000                  4$:     0                       ;ERROR RES.
22361  114444  000000                          0
22362  114446  000000                          0
22363  114450  000000                          0
22364  114452  000200                  5$:     200                     ;FPS BEFORE EXECUTION.
22365  114454  000204                          204                     ;FPS AFTER EXECUTION.
22366  114456  000214                          214                     ;ERROR FPS.
22367  114460  177777                          -1
22368                                  ;TEST TSTD -0 OP FIUV=0
22369  114462  004767  000114          WMB10:  JSR     PC,NATSUB
22370  114466  000002                  1$:     2                       ;FLAG=TSTD.
22371  114470  100123                  2$:     100123                  ;OPERAND.
22372  114472  021012                          21012
22373  114474  034565                          34565
22374  114476  043210                          43210
22375  114500  100123                  3$:     100123                  ;RESULT.
22376  114502  021012                          21012
22377  114504  034565                          34565
22378  114506  043210                          43210
22379  114510  000000                  4$:     0                       ;ERROR RES.
22380  114512  000000                          0
22381  114514  000000                          0
22382  114516  000000                          0
22383  114520  040203                  5$:     40203                   ;FPS BEFORE EXECUTION.
22384  114522  040214                          040214                  ;FPS AFTER EXECUTION.
22385  114524  140214                          140214                  ;ERROR FPS.
22386  114526  177777                          -1
22387                                  ;TEST TSTD -0 OP FIUV=1
22388  114530  004767  000046          WMB11:  JSR     PC,NATSUB
22389  114534  000002                  1$:     2                       ;FLAG=TSTD.
22390  114536  100137                  2$:     100137                  ;OPERAND.
22391  114540  024613                          24613
22392  114542  057024                          57024
22393  114544  060137                          60137
22394  114546  100137                  3$:     100137                  ;RESULT.
22395  114550  024613                          24613
22396  114552  057024                          57024
22397  114554  060137                          60137
22398  114556  000000                  4$:     0                       ;ERROR RES.
22399  114560  000000                          0
22400  114562  000000                          0
22401  114564  000000                          0
22402  114566  044200                  5$:     44200                   ;FPS BEFORE EXECUTION.
22403  114570  144214                          144214                  ;FPS AFTER EXECUTION.
22404  114572  044214                          044214                  ;ERROR FPS.
22405  114574  000014                          14
22406  114576  000167  000162                  JMP     WMBDONE
22407
22408                                  ;THIS SUBROUTINE, NATSUB, IS USED TO SET UP THE OPERANDS, EXECUTE
```

```
                        ;THE EITHER A TSTD, AN ABSD OR A NEGD INSTRUCTION AND CHECK THE RESULTS. A CALL
                        ;TO IT IS MADE THUS:
                        ;
                        ;               JSR     PC,@#NATSUB
                        ;               FLAG:   .WORD   X               ;INSTRUCTION TYPE FLAG.
                        ;               ACARG:  .WORD   X,X,X,X         ;OPERAND
                        ;               RES:    .WORD   X,X,X,X         ;EXPECTED RESULT
                        ;               ERRES:  .WORD   X,X,X,X         ;ERROR RESULT
                        ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
                        ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
                        ;               FEC:    .WORD   X               ;EXPECTED FEC
                        ;               ERFPS:  .WORD   X               ;ERROR FPS.
                        ;               ERR1:   ERROR   X               ;DATA ERROR.
                        ;                       BR      CONT
                        ;               ERR2:   ERROR   X               ;FPS ERROR.
                        ;               CONT:                           ;RETURN ADDRESS
                        ;
                        ;THE OPERAND IS SET UP IN NATBF1. THEN
                        ;THE EITHER THE TSTD, NEGD OR ABSD INSTRUCTION IS EXECUTED.
                        ;NATSUB USES THE FIRST OPERAND AS A FLAG TO DETERMINE WHICH INSTRUCTION
                        ;IS TO BE EXECUTED: 0 = NEGD, 1 = ABSD, 2 = TSTD.
                        ;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
                        ;COMPARED WITH FPSA. IF THIS TOO IS CORRECT NATSUB RETURNS CONTROL
                        ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD NATSUB
                        ;COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN NATSUB WILL RETURN
                        ;TO THE ERROR CALL AT ERR2, OTHERWISE NATSUB ITSELF
                        ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
                        ;INSTRUCTION IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
                        ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
                        ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN NATSUB
                        ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
                        ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND NATSUB WILL
                        ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.


114602  012601          NATSUB: MOV     (SP)+,R1                ;GET A POINTER TO THE ARGUMENTS.
114604  010102                  MOV     R1,R2           ;COPY THE OPERAND.
114606  062702  000002          ADD     #2,R2
114612  012703  114752          MOV     #NATBF1,R3
114616  012704  000004          MOV     #4,R4
114622  012223          1$:     MOV     (R2)+,(R3)+
114624  077402                  SOB     R4,1$
114626  016100  000032          MOV     32(R1),R0       ;LOAD THE FPS.
114632  170100                  LDFPS   R0
114634  012700  114752          MOV     #NATBF1,R0      ;SET UP THE OPERAND ADDRESS.
114640  011102                  MOV     (R1),R2         ;GET THE FLAG TO DETERMINE WHICH
114642  006302                  ASL     R2              ;INSTRUCTION TO EXECUTE.
114644  006302                  ASL     R2              ;0 = NEGD, 1 = ABSD, 2 = TSTD
114646  012703  114656          MOV     #NATINS,R3
114652  060203                  ADD     R2,R3
114654  000113                  JMP     (R3)            ;GO EXECUTE THE INSTRUCTION.
114656  170710          NATINS: NEGD    (R0)
114660  000403                  BR      2$
114662  170610                  ABSD    (R0)
114664  000401                  BR      2$
114666  170510                  TSTD    (R0)
```

```
22465
22466  114670  170204                    2$:     STFPS   R4              ;GET THE FPS.
22467  114672  170305                            STST    R5              ;GET THE FEC.
22468  114674  010100                            MOV     R1,R0           ;WAS THE RESULT CORRECT?
22469  114676  062700  000012                    ADD     #12,R0
22470  114702  012702  114752                    MOV     #NATBF1,R2
22471  114706  012703  000004                    MOV     #4,R3
22472  114712  022022                    3$:     CMP     (R0)+,(R2)+
22473  114714  001014                            BNE     10$             ;BRANCH IF INCORRECT.
22474  114716  077303                            SOB     R3,3$
22475  114720  026104  000034                    CMP     34(R1),R4       ;WAS THE FPS CORRECT?
22476  114724  001010                            BNE     10$             ;BRANCH IF INCORRECT.
22477  114726  005761  000034                    TST     34(R1)          ;IF THE EXPECTED FPS WAS NEGATIVE CHECK THE FEC.
22478  114732  100003                            BPL     4$
22479  114734  026105  000040                    CMP     40(R1),R5       ;WAS THE FEC CORRECT.
22480  114740  001002                            BNE     10$             ;BRANCH IF INCORRECT.
22481  114742  000161  000042            4$:     JMP     42(R1)          ;RETURN.
22482
22483  114746                            10$:
  (2)  114746  104000                            EMT                     ;
22484
22485  114750  177777                            .WORD   -1
22486  114752  177777  177777  177777  NATBF1: .WORD   -1,-1,-1,-1,-1
       114760  177777  177777
22487
22488  114764                            WWBDONE:
  (1)  114764  004767  007562                    JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                            ;SEE IF THE USER HAS EXPRESSED
  (1)                                            ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                            ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                            ;THE USER TYPED CONTROL G?).
22489
22490
22497
22498                               ;*****************************************************************************
  (2)                               ;TEST 556       SOURCE MODES, MODE 1 (FL=0), TEST
  (3)                               ;*****************************************************************************
  (2)  114770                       TS556:
22499
22500
22501
22502  114770  012700  115040                    MOV     #AACTP1,R0      ;SET UP TEST DATA IN BUFFER.
22503  114774  012710  147517                    MOV     #147517,(R0)
22504  115000  012737  115014  037576            MOV     #AAC2,@#TMP2
22505  115006  012737  115044  000004            MOV     #AAC10,@#ERRVECT        ;SET UP FOR TRAPS TO 4.
22506  115014  170110                    AAC2:   LDFPS   (R0)            ;TEST INSTRUCTION.
22507
22508  115016  170205                            STFPS   R5              ;GET FPS
22509
22510  115020  020027  115040                    CMP     R0,#AACTP1      ;IS R0 CORRECT?
22511  115024  001007                            BNE     AAC10           ;BR IF NOT.
22512  115026  022705  147517                    CMP     #147517,R5      ;IS FPS CORRECT?
22513  115032  001004                            BNE     AAC10           ;BR IF NOT.
22514  115034  000404                            BR      AACDONE
22515
22516                               ;TEST BUFFER AND DATA:
```

```
22517  115036  177777                          -1
22518  115040  147517                AACTP1:   147517
22519  115042  177777                          -1
22520  115044                        AAC10:
  (2)  115044  104000                          EMT                             ;
22521
22522  115046                        AACDONE:
  (1)  115046  004767  007500                  JSR     PC,..RSET               ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                          ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                          ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                          ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                          ;THE USER TYPED CONTROL G?).
22523
22524
22525                               ;*************************************************************************************
  (2)                               ;TEST 557       SOURCE MODES, MODE 2 (FL=0), TEST
  (3)                               ;*************************************************************************************
  (2)  115052                       TS557:
22526
22527
22528  115052  012700  115114                   MOV     #BBCTP1,R0              ;SET UP TEST DATA IN BUFFER.
22529  115056  012710  145212                   MOV     #145212,(R0)
22530  115062  012737  115120  000004           MOV     #BBC10,@#ERRVECT        ;SET UP FOR TRAPS TO 4.
22531
22532  115070  170120                BBC2:     LDFPS   (R0)+                   ;TEST INSTRUCTION.
22533
22534  115072  170205                          STFPS   R5                      ;GET FPS
22535
22536  115074  020027  115116                   CMP     R0,@#BBCTP1+2           ;IS R0 CORRECT?
22537  115100  001007                          BNE     BBC10                   ;BR IF NOT.
22538  115102  022705  145212                   CMP     #145212,R5              ;IS THE FPS CORRECT?
22539  115106  001004                          BNE     BBC10                   ;BR IF NOT.
22540  115110  000404                          BR      BBCDONE
22541
22542
22543                             ;TEST BUFFER AND DATA:
22544  115112  177777                          -1
22545  115114  177777                BBCTP1:   .WORD   -1
22546  115116  177777                          -1
22547
```

```
22549
22550  115120                          BBC10:
  (2)  115120  104000                          EMT                     ;
22551  115122                          BBCDONE:
  (1)  115122  004767  007424                  JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
22552
22553
22554                                  ;*************************************************************************
  (2)                                  ;TEST 560       SOURCE MODES, MODE 4 (FL=0), TEST
  (3)                                  ;*************************************************************************
  (2)  115126                          TS560:
22555
22556
22557  115126  012700  115210                  MOV     #DDCTP1+2,R0    ;SET UP THE TEST DATA BUFFER.
22558  115132  012760  105252  177776          MOV     #105252,-2(R0)
22559  115140  012737  115154  037576          MOV     #DDC2,@#TMP2
22560  115146  012737  115220  000004          MOV     #DDC10,@#ERRVEC
22561  115154  170140                  DDC2:   LDFPS   -(R0)
22562  115156  170205                          STFPS   R5
22563  115160  020027  115206                  CMP     R0,#DDCTP1
22564  115164  001015                          BNE     DDC10
22565  115166  022705  105252                  CMP     #105252,R5
22566  115172  001012                          BNE     DDC10
22567  115174  000412                          BR      DDCDONE
22568
22569  115176  177777  177777  177777          -1,-1,-1,-1
       115204  177777
22570  115206  177777                  DDCTP1: -1
22571  115210  177777  177777  177777          -1,-1,-1,-1
       115216  177777
22572  115220                          DDC10:
  (2)  115220  104000                          EMT                     ;
22573  115222                          DDCDONE:
  (1)  115222  004767  007324                  JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
22574                                  ;*************************************************************************
  (2)                                  ;TEST 561       SOURCE MODES, MODE 3 (FL=0), TEST
  (3)                                  ;*************************************************************************
  (2)  115226                          TS561:
22575  115226  012700  115314                  MOV     #EECTP2,R0
22576  115232  012710  115304                  MOV     #EECTP1,(R0)
22577  115236  012767  103456  000040          MOV     #103456,EECTP1
22578  115244  012737  115326  000004          MOV     #EEC10,@#ERRVECT ;SET UP FOR TRAPS TO 4.
22579  115252  170130                  EEC2:   LDFPS   @(R0)+          ;TEST INSTRUCTION.
22580  115254  170205                          STFPS   R5              ;GET THE FPS.
22581  115256  020027  115316                  CMP     R0,#EECTP2+2    ;IS R0 CORRECT?
22582  115262  001021                          BNE     EEC10           ;BR IF NOT.
22583  115264  022705  103456                  CMP     #103456,R5      ;IS THE FPS CORRECT?
22584  115270  001016                          BNE     EEC10           ;BR IF NOT.
```

```
22585  115272  000416                          BR      EECDONE
22586
22587
22588                                  ;TEST BUFFER AND DATA:
22589  115274  177777  177777  177777          -1,-1,-1,-1
       115302  177777
22590  115304  177777                  EECTP1: -1
22591  115306  177777  177777  177777          -1,-1,-1
22592  115314  115304  177777  177777  EECTP2: EECTP1,-1,-1,-1,
       115322  177777  000000
22593
22594  115326                          EEC10:
  (2)  115326  104000                          EMT                     ;
22595  115330                          EECDONE:
  (1)  115330  004767  007216                  JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
22596                                  ;*************************************************************************
  (2)                                  ;TEST 562       SOURCE MODES, MODE 5 (FL=0), TEST
  (3)                                  ;*************************************************************************
  (2)  115334                          TS562:
22597  115334  012700  115420                  MOV     #FFCTP2+2,R0    ;SET UP THE TEST DATA BUFFER.
22598  115340  012760  115406  177776          MOV     #FFCTP1,-2(R0)
22599  115346  012737  045412  115406          MOV     #45412,@#FFCTP1
22600  115354  012737  115426  000004          MOV     #FFC10,@#ERRVECT ;SET UP FOR TRAPS TO 4.
22601  115362  170150                  FFC2:   LDFPS   @-(R0)          ;TEST INSTRUCTION.
22602  115364  170205                          STFPS   R5              ;GET THE FPS.
22603  115366  020027  115416                  CMP     R0,#FFCTP2      ;IS R0 CORRECT?
22604  115372  001015                          BNE     FFC10           ;BR IF NOT.
22605  115374  022705  045412                  CMP     #45412,R5       ;IS THE FPS CORRECT?
22606  115400  001012                          BNE     FFC10           ;BR IF NOT.
22607  115402  000412                          BR      FFCDONE
22608
22609
22610                                  ;TEST BUFFER AND DATA:
22611  115404  177777                          -1
22612  115406  177777                  FFCTP1: -1
22613  115410  177777  177777  177777          -1,-1,-1
22614  115416  115406  177777  177777  FFCTP2: FFCTP1,-1,-1,-1
       115424  177777
22615
22616  115426                          FFC10:
  (2)  115426  104000                          EMT                     ;
22617  115430                          FFCDONE:
  (1)  115430  004767  007116                  JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
22618                                  ;*************************************************************************
  (2)                                  ;TEST 563       SOURCE MODES, MODE 6 (FL=0), TEST
  (3)                                  ;*************************************************************************
  (2)  115434                          TS563:
22619  115434  012700  110307                  MOV     #GGCTP1-5201,R0 ;SET UP THE TEST DATA BUFFER.
```

```
 22620  115440  012737  046543  115510          MOV     #46543,@#GGCTP1
 22621  115446  005001                          CLR     R1
 22622  115450  012737  115522  000004          MOV     #GGC10,@#ERRVECT ;SET UP FOR TRAPS TO 4.
 22623  115456  170160  005201          GGC2:   LDFPS   5201(R0)                 ;TEST INSTRUCTION.
 22624  115462  170204                          STFPS   R4               ;GET THE FPS.
 22625  115464  005701                          TST     R1               ;WAS PC CORRECT AFTER EXECUTION?
 22626  115466  001015                          BNE     GGC10            ;BR IF NOT.
 22627  115470  020027  110307                  CMP     R0,#GGCTP1-5201  ;IS R0 CORRECT?
 22628  115474  001012                          BNE     GGC10            ;BR IF NOT.
 22629  115476  022704  046543                  CMP     #46543,R4        ;IS THE FPS CORRECT?
 22630  115502  001007                          BNE     GGC10            ;BR IF NOT.
 22631  115504  000407                          BR      GGCDONE
 22632
 22633
 22634                                  ;TEST BUFFER AND DATA:
 22635  115506  177777                            -1
 22636  115510  177777  177777  177777  GGCTP1: -1,-1,-1,-1
        115516  177777
 22637  115520  177777                            -1
 22638  115522                          GGC10:
   (2)  115522  104000                          EMT                      ;
 22639  115524                          GGCDONE:
   (1)  115524  004767  007022                  JSR     PC,.RSET         ;GO INITIALIZE THE FPS AND STACK; AND
   (1)                                                                   ;SEE IF THE USER HAS EXPRESSED
   (1)                                                                   ;THE DESIRE TO CHANGE THE SOFTWARE
   (1)                                                                   ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
   (1)                                                                   ;THE USER TYPED CONTROL G?).
 22640                                  ;*************************************************************************
   (2)                                  ;TEST 564       SOURCE MODES, MODE 7 (FL=0), TEST
   (3)                                  ;*************************************************************************
   (2)  115530                          TS564:
 22641  115530  012700  110421                  MOV     #HHCTP2-5201,R0 ;SET UP THE TEST DATA BUFFER.
 22642  115534  012760  115612  005201          MOV     #HHCTP1,5201(R0)
 22643  115542  012737  004547  115612          MOV     #4547,@#HHCTP1
 22644  115550  005001                          CLR     R1
 22645  115552  012737  115632  000004          MOV     #HHC10,@#ERRVECT ;SET UP FOR TRAPS TO 4.
 22646  115560  170170  005201          HHC2:   LDFPS   @5201(R0)                ;TEST INSTRUCTION.
 22647  115564  170204                          STFPS   R4               ;GET THE FPS.
 22648  115566  005701                          TST     R1               ;WAS PC CORRECT AFTER EXECUTION?
 22649  115570  001020                          BNE     HHC10            ;BR IF NOT.
 22650  115572  020027  110421                  CMP     R0,#HHCTP2-5201  ;IS R0 CORRECT?
 22651  115576  001015                          BNE     HHC10            ;BR IF NOT.
 22652  115600  022704  004547                  CMP     #4547,R4         ;IS THE FPS CORRECT?
 22653  115604  001012                          BNE     HHC10            ;BR IF NOT.
 22654  115606  000412                          BR      HHCDONE
 22655
 22656
 22657                                  ;TEST BUFFER AND DATA:
 22658  115610  177777                            -1
 22659  115612  177777  177777  177777  HHCTP1: .WORD -1,-1,-1,-1
        115620  177777
 22660  115622  177777  177777  177777  HHCTP2: .WORD -1,-1,-1,-1
        115630  177777
 22661  115632                          HHC10:
   (2)  115632  104000                          EMT                      ;
 22662  115634                          HHCDONE:
```

```
  (1)  115634  004767  006712                JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
22663
22664
22671
22672                                ;*************************************************************************************
  (2)                                ;TEST 565       SOURCE MODES, MODE 2 GR7 (FL=1), TEST
  (3)                                ;*************************************************************************************
  (2)  115640                        TS565:
22673
22674  115640  012737  115676  000004        MOV     #IIC20,@#ERRVECT ;SET UP FOR TRAPS TO 4.
22675  115646  012700  000300                MOV     #300,R0
22676  115652  170100                        LDFPS   R0
22677  115654  005001                        CLR     R1
22678
22679  115656  177027                IIC2:   LDCLD   (R7)+,AC0               ;TEST INSTRUCTION.
22680  115660  005201                        5201
22681  115662  005201                        5201
22682  115664  005201                        5201
22683  115666  005201                        5201
22684
22685  115670  020127  000003                CMP     R1,#3           ;WAS PC CORRECT AFTER EXECUTION?
22686  115674  001401                        BEQ     IICDONE
  (1)  115676                        IIC20:
  (2)  115676  104000                        EMT                     ;
22687
22688  115700                        IICDONE:
  (1)  115700  004767  006646                JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
22689
22696
22697                                ;*************************************************************************************
  (2)                                ;TEST 566       SOURCE MODES, MODE 2 (FL=1), TEST
  (3)                                ;*************************************************************************************
  (2)  115704                        TS566:
22698
22699  115704  012700  000300                MOV     #300,R0
22700  115710  170100                        LDFPS   R0
22701  115712  012700  115756                MOV     #TCCBF0,R0      ;SET UP THE TEST DATA BUFFER.
22702  115716  177020                TCC2:   LDCLD   (R0)+,AC0               ;TEST INSTRUCTION.
22703
22704  115720  170204                        STFPS   R4              ;GET THE FPS.
22705  115722  012701  115766                MOV     #TCCBF1,R1      ;GET THE RESULT.
22706  115726  012702  000200                MOV     #200,R2
22707  115732  170102                        LDFPS   R2
22708  115734  174011                        STD     AC0,(R1)
22709  115736  020027  115762                CMP     R0,#TCCBF0+4    ;IS R0 CORRECT?
22710  115742  001401                        BEQ     TCC3
  (2)  115744  104000                        EMT                     ;
22711
```

```
22712  115746  022704  000300          TCC3:   CMP     #300,R4 ;IS THE FPS CORRECT?
22713  115752  001411                  BEQ     TCCDONE
  (2)  115754  104000                  EMT                     ;
22714
22715
22716                                  ;TEST BUFFER AND DATA:
22717  115756  001234  067076  054321  TCCBF0: .WORD   01234,67076,54321,012345
       115764  012345
22718  115766  177777  177777  177777  TCCBF1: -1,-1,-1,-1
       115774  177777
22719
22720  115776                          TCCDONE:
  (1)  115776  004767  006550                  JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
22721
22722
22729
22730                                  ;*************************************************************************
  (2)                                  ;TEST 567       LDCIF AND LDCLF TEST
  (3)                                  ;*************************************************************************
  (2)  116002                          TS567:
22731
22732
22733                                  ;ZERO   OPERAND FL=0
22734
22735  116002  004737  116674          KKC1:   JSR     PC,@#LDCFSUB    ;GO EXECUTE INSTRUCTION.
22736
22737  116006  000000  000000          1$:     .WORD   0,0             ;FSRC OPERAND.
22738  116012  000000  000000          2$:     .WORD   0,0             ;EXPECTED RESULT.
22739  116016  177777  177777          3$:     .WORD   -1,-1           ;ANTICIPATED ERRONEOUS RESULT.
22740  116022  000000                  4$:     0                       ;FPS BEFORE EXECUTION.
22741  116024  000004                          4                       ;FPS AFTER EXECUTION.
22742  116026  177777                          -1                      ;ANTICIPATED ERRONEOUS FPS.
22743                                  ;ZERO   OPERAND FL=0
22744
22745  116030  004737  116674          KKC2:   JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
22746
22747  116034  000000  177777          1$:     .WORD   0,-1            ;FSRC OPERAND.
22748  116040  000000  000000          2$:     .WORD   0,0             ;EXPECTED RESULT.
22749  116044  004177  177400          3$:     4177,177400             ;ANTICIPATED ERRONEOUS RESULT.
22750  116050  000000                  4$:     0                       ;FPS BEFORE EXECUTION.
22751  116052  000004                          4                       ;FPS AFTER EXECUTION.
22752  116054  177777                          -1                      ;ANTICIPATED ERRONEOUS FPS.
22753                                  ;ZERO   OPERAND FL=1
22754
22755  116056  004737  116674          KKC3:   JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
22756
22757  116062  000000  000000          1$:     .WORD   0,0             ;FSRC OPERAND.
22758  116066  000000  000000          2$:     .WORD   0,0             ;EXPECTED RESULT.
22759  116072  177777  177777          3$:     .WORD   -1,-1           ;ANTICIPATED ERRONEOUS RESULT.
22760  116076  000100                  4$:     100                     ;FPS BEFORE EXECUTION.
22761  116100  000104                          104                     ;FPS AFTER EXECUTION.
22762  116102  000004                          4                       ;ANTICIPATED ERRONEOUS FPS.
```

```
22763                                   ;OPERAND        POSITIVE        FL=0
22764  116104  004737  116674           KKC4:   JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
22765  116110  040000  000000           1$:     .WORD   40000.0         ;FSRC OPERAND.
22766  116114  043600  000000           2$:     .WORD   43600.0         ;EXPECTED RESULT.
22767  116120  047600  000000           3$:     .WORD   47600.0         ;ANTICIPATED ERRONEOUS RESULT.
22768  116124  000017                   4$:     17                      ;FPS BEFORE EXECUTION.
22769  116126  000000                           0                       ;FPS AFTER EXECUTION.
22770  116130  177777                           -1                      ;ANTICIPATED ERRONEOUS FPS.
22771                                   ;OPERAND=1.     FL=0
22772  116132  004737  116674           KKC5:   JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
22773  116136  000001  000000           1$:     .WORD   1.0             ;FSRC OPERAND.
22774  116142  040200  000000           2$:     .WORD   40200.0         ;EXPECTED RESULT.
22775  116146  044200  000000           3$:     .WORD   44200.0         ;ANTICIPATED ERRONEOUS RESULT.
22776  116152  000017                   4$:     17                      ;FPS BEFORE EXECUTION.
22777  116154  000000                           0                       ;FPS AFTER EXECUTION.
22778  116156  177777                           -1                      ;ANTICIPATED ERRONEOUS FPS.
22779                                   ;OPERAND=       PATTERN FL=0
22780  116160  004737  116674           KKC6:   JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
22781  116164  000252  000000           1$:     .WORD   252.0           ;FSRC OPERAND.
22782  116170  042052  000000           2$:     .WORD   42052.0         ;EXPECTED RESULT.
22783  116174  046052  000000           3$:     .WORD   46052.0         ;ANTICIPATED ERRONEOUS RESULT.
22784  116200  000000                   4$:     0                       ;FPS BEFORE EXECUTION.
22785  116202  000000                           0                       ;FPS AFTER EXECUTION.
22786  116204  177777                           -1                      ;ANTICIPATED ERRONEOUS FPS.
22787                                   ;OPERAND=-40000 FL=0
22788  116206  004737  116674           KKC7:   JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
22789  116212  140000  000000           1$:     .WORD   -40000.0                ;FSRC OPERAND.
22790  116216  143600  000000           2$:     .WORD   143600.0                ;EXPECTED RESULT.
22791  116222  043600  000000           3$:     .WORD   43600.0         ;ANTICIPATED ERRONEOUS RESULT.
22792  116226  000007                   4$:     7                       ;FPS BEFORE EXECUTION.
22793  116230  000010                           10                      ;FPS AFTER EXECUTION.
22794  116232  177777                           -1                      ;ANTICIPATED ERRONEOUS FPS.
22795                                   ;OPERAND=-1     FL-0
22796  116234  004737  116674           KKC10:  JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
22797  116240  177777  000000           1$:     .WORD   -1.0            ;FSRC OPERAND.
22798  116244  140200  000000           2$:     .WORD   140200.C                ;EXPECTED RESULT.
22799  116250  144000  000400           3$:     .WORD   144000.400              ;ANTICIPATED ERRONEOUS RESULT.
22800  116254  000000                   4$:     0                       ;FPS BEFORE EXECUTION.
22801  116256  000010                           10                      ;FPS AFTER EXECUTION.
22802  116260  177777                           -1                      ;ANTICIPATED ERRONEOUS FPS.
22803                                   ;OPERAND=PATTERN        FL=0
22804  116262  004737  116674           KKC11:  JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
22805  116266  125252  000000           1$:     .WORD   125252.0                ;FSRC OPERAND.
22806  116272  143652  126000           2$:     .WORD   143652.126000           ;EXPECTED RESULT.
22807  116276  043652  126000           3$:     .WORD   43652.126000            ;ANTICIPATED ERRONEOUS RESULT.
22808  116302  000007                   4$:     7                       ;FPS BEFORE EXECUTION.
22809  116304  000010                           10                      ;FPS AFTER EXECUTION.
22810  116306  177777                           -1                      ;ANTICIPATED ERRONEOUS FPS.
22811                                   ;OPERAND        POS     FL-1
22812  116310  004737  116674           KKC12:  JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
22813  116314  040000  000000           1$:     .WORD   40000.0         ;FSRC OPERAND.
22814  116320  047600  000000           2$:     .WORD   47600.0         ;EXPECTED RESULT.
22815  116324  043600  000000           3$:     .WORD   43600.0         ;ANTICIPATED ERRONEOUS RESULT.
22816  116330  000117                   4$:     117                     ;FPS BEFORE EXECUTION.
22817  116332  000100                           100                     ;FPS AFTER EXECUTION.
22818  116334  177777                           -1                      ;ANTICIPATED ERRONEOUS FPS.
```

```
22819
22820  116336  004737  116674          ;OPERAND=1      FL=1
KKC13:  JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
1$:     .WORD   0,1             ;FSRC OPERAND.
2$:     .WORD   40200,0         ;EXPECTED RESULT.
3$:     .WORD   34200,0         ;ANTICIPATED ERRONEOUS RESULT.
4$:     100                     ;FPS BEFORE EXECUTION.
        100                     ;FPS AFTER EXECUTION.
        -1                      ;ANTICIPATED ERRONEOUS FPS.
22827          ;OPERAND=        PATTERN FL=1
KKC14:  JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
1$:     .WORD   0,252           ;FSRC OPERAND.
2$:     .WORD   42052,0         ;EXPECTED RESULT.
3$:     .WORD   36052,0         ;ANTICIPATED ERRONEOUS RESULT.
4$:     111                     ;FPS BEFORE EXECUTION.
        100                     ;FPS AFTER EXECUTION.
        -1                      ;ANTICIPATED ERRONEOUS FPS.
22835          ;OPERAND=-40000,0       FL=1
KKC15:  JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
1$:     .WORD   -40000,0                ;FSRC OPERAND.
2$:     .WORD   147600,0                ;EXPECTED RESULT.
3$:     .WORD   47600,0         ;ANTICIPATED ERRONEOUS RESULT.
4$:     107                     ;FPS BEFORE EXECUTION.
        110                     ;FPS AFTER EXECUTION.
        -1                      ;ANTICIPATED ERRONEOUS FPS.
22843          ;OPERAND=-1,-1  FL=1
KKC16:  JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
1$:     .WORD   -1,-1           ;FSRC OPERAND.
2$:     .WORD   140200,0                ;EXPECTED RESULT.
3$:     .WORD   150000,0                ;ANTICIPATED ERRONEOUS RESULT.
4$:     100             ;FPS BEFORE EXECUTION.
        110             ;FPS AFTER EXECUTION.
        -1              ;ANTICIPATED ERRONEOUS FPS.
22851          ;OPERAND=-PATTERN       FL=1,   ROUND MODE
KKC17:  JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
1$:     .WORD   125252,125252           ;FSRC OPERAND.
2$:     .WORD   147652,125253           ;EXPECTED RESULT.
3$:     .WORD   47652,125253            ;ANTICIPATED ERRONEOUS RESULT.
4$:     105             ;FPS BEFORE EXECUTION.
        110             ;FPS AFTER EXECUTION.
        -1              ;ANTICIPATED ERRONEOUS FPS.
22859          ;OPERAND=77777,177500   FL=1,   ROUND MODE
KKC20:  JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
1$:     .WORD   77777,177500            ;FSRC OPERAND.
2$:     .WORD   47777,177777            ;EXPECTED RESULT.
3$:     .WORD   47777,177776            ;ANTICIPATED ERRONEOUS RESULT.
4$:     117             ;FPS BEFORE EXECUTION.
        100             ;FPS AFTER EXECUTION.
        -1              ;ANTICIPATED ERRONEOUS FPS.
22867          ;OPERAND=40000,000100   FL=1,   ROUND MODE
KKC21:  JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
1$:     .WORD   40000,100               ;FSRC OPERAND.
2$:     .WORD   47600,1         ;EXPECTED RESULT.
3$:     .WORD   47600,0         ;ANTICIPATED ERRONEOUS RESULT.
4$:     102             ;FPS BEFORE EXECUTION.
        100             ;FPS AFTER EXECUTION.
        -1              ;ANTICIPATED ERRONEOUS FPS.
```

```
22875                                   ;OPERAND=40000,000100   FL=1,    TRUNC MODE
22876  116570  004737  116674   KKC22:  JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
22877  116574  040000  000100   1$:     .WORD   40000,100               ;FSRC OPERAND.
22878  116600  047600  000000   2$:     .WORD   47600,0         ;EXPECTED RESULT.
22879  116604  047600  000001   3$:     .WORD   47600,1         ;ANTICIPATED ERRONEOUS RESULT.
22880  116610  000157           4$:     157                     ;FPS BEFORE EXECUTION.
22881  116612  000140                   140                     ;FPS AFTER EXECUTION.
22882  116614  177777                   -1                      ;ANTICIPATED ERRONEOUS FPS.
22883                                   ;OPERAND=100000,0 (MOST NEG #)  FL=0
22884  116616  004737  116674   KKC23:  JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
22885  116622  100000  000000   1$:     .WORD   100000,0                ;FSRC OPERAND.
22886  116626  144000  000000   2$:     .WORD   144000,0                ;EXPECTED RESULT.
22887  116632  143600  000000   3$:     .WORD   143600,0                ;ANTICIPATED ERRONEOUS RESULT.
22888  116636  000007           4$:     7                       ;FPS BEFORE EXECUTION.
22889  116640  000010                   10                      ;FPS AFTER EXECUTION.
22890  116642  177777                   -1                      ;ANTICIPATED ERRONEOUS FPS.
22891                                   ;OPERAND=100000,0       FL=1
22892  116644  004737  116674   KKC24:  JSR     PC,@#LDCFSUB    ;GO EXECUTE THE INSTRUCTION.
22893  116650  100000  000000   1$:     .WORD   100000,0                ;FSRC OPERAND.
22894  116654  150000  000000   2$:     .WORD   150000,0                ;EXPECTED RESULT.
22895  116660  147600  000000   3$:     .WORD   147600,0                ;ANTICIPATED ERRONEOUS RESULT.
22896  116664  000107           4$:     107                     ;FPS BEFORE EXECUTION.
22897  116666  000110                   110                     ;FPS AFTER EXECUTION.
22898  116670  177777                   -1                      ;ANTICIPATED ERRONEOUS FPS.
22899  116672  000441           6$:     BR      KKCDONE
22900
22901                                   ;THIS SUBROUTINE, LDCFSUB, IS USED TO SET UP THE OPERANDS, EXECUTE
22902                                   ;THE LDCIF OR LDCLF INSTRUCTION AND CHECK THE RESULTS. A CALL
22903                                   ;TO IT IS MADE THUS:
22904                                   ;
22905                                   ;               JSR     PC,@#LDCFSUB
22906                                   ;               ACARG:  .WORD   X,X             ;AC OPERAND
22907                                   ;               RES:    .WORD   X,X             ;EXPECTED RESULT
22908                                   ;               ERRES:  .WORD   X,X             ;ERROR RESULT
22909                                   ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
22910                                   ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
22911                                   ;               ERFPS:  .WORD   X               ;ERROR FPS
22912                                   ;               ERR1:   ERROR   X               ;DATA ERROR
22913                                   ;                       BR      CONT
22914                                   ;               ERR2:   ERROR   X               ;FPS ERROR
22915                                   ;               CONT:                           ;RETURN ADDRESS
22916                                   ;
22917                                   ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
22918                                   ;THE LDCIF OR LDCLF INSTRUCTION IS EXECUTED.
22919                                   ;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
22920                                   ;COMPARED WITH FPSA IF THIS TOO IS CORRECT LDCFSUB RETURNS CONTROL
22921                                   ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD LDCFSUB WILL
22922                                   ;COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN LDCFSUB WILL RETURN
22923                                   ;TO THE ERROR CALL AT ERR2, OTHERWISE LDCFSUB ITSELF
22924                                   ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
22925                                   ;LDCIF OR LDCLF IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
22926                                   ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
22927                                   ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN LDCFSUB
22928                                   ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
22929                                   ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND LDCFSUB
22930                                   ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.
```

```
22931
22932  116674  012601                  LDCFSUB:        MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
22933  116676  016100  000014                  MOV     14(R1),R0       ;SET THE FPS.
22934  116702  170100                  LDFPS   R0
22935  116704  010100                  MOV     R1,R0
22936
22937  116706  177010                  1$:     LDCIF   (R0),AC0                ;TEST INSTRUCTION LDCIF OR LDCLF.
22938
22939  116710  170204                  STFPS   R4              ;GET FPS.
22940  116712  012700  116766          MOV     #LDCT,R0        ;GET THE RESULT.
22941  116716  012702  000200          MOV     #200,R2
22942  116722  170102                  LDFPS   R2
22943  116724  174010                  STD     AC0,(R0)
22944
22945  116726  012702  116766          MOV     #LDCT,R2        ;SEE IF THE RESULT WAS CORRECT.
22946  116732  010100                  MOV     R1,R0
22947  116734  062700  000004          ADD     #4,R0
22948  116740  012703  000002          MOV     #2,R3
22949  116744  022022                  2$:     CMP     (R0)+,(R2)+
22950  116746  001006                  BNE     10$             ;BR IF INCORRECT.
22951  116750  077303                  SOB     R3,2$
22952
22953  116752  026104  000016          CMP     16(R1),R4       ;SEE IF THE FPS WAS CORRECT.
22954  116756  001002                  BNE     10$             ;BR IF INCORRECT.
22955  116760  000161  000022          3$:     JMP     22(R1)          ;RETURN.
22956  116764                          10$:
  (2)  116764  104000                  EMT                     ;
22957
22958                                  ;DATA BUFFER:
22959  116766  000000  000000  000000  LDCT:   .WORD   0,0,0,0
       116774  000000
22960
22961  116776                          KKCDONE:
  (1)  116776  004767  005550                  JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
22962
22968
22969                                  ;*************************************************************************************
  (2)                                  ;TEST 570       LDCID AND LDCLD TEST
  (3)                                  ;*************************************************************************************
  (2)  117002                          TS570:
22970                                  ;OPERAND=0      FL=0,   FD=1
22971  117002  004737  117460          LLC1:   JSR     PC,@#LDCDSUB    ;GO EXECUTE THE INSTRUCTION.
22972  117006  000000  000000          1$:     .WORD   0,0             ;FSRC OPERAND.
22973  117012  000000  000000  000000  2$:     .WORD   0,0,0,0         ;EXPECTED RESULT.
       117020  000000
22974  117022  177777  177777  177777  3$:     .WORD   -1,-1,-1,-1             ;ANTICIPATED ERRONEOUS RESULT.
       117030  177777
22975  117032  000213                  4$:     213                     ;FPS BEFORE EXECUTION.
22976  117034  000204                          204                     ;FPS AFTER EXECUTION.
22977  117036  177777                          -1                      ;ANTICIPATED ERRONEOUS FPS.
22978                                  ;OPERAND=0      FL=0,   FD=1
22979  117040  004737  117460          LLC2:   JSR     PC,@#LDCDSUB    ;GO EXECUTE THE INSTRUCTION.
```

```
22980  117044  000000  177777          1$:     .WORD   0,-1                    ;FSRC OPERAND.
22981  117050  000000  000000  000000  2$:     .WORD   0,0,0,0                 ;EXPECTED RESULT.
       117056  000000
22982  117060  004177  177400  000000  3$:     .WORD   4177,177400,0,0                 ;ANTICIPATED ERRONEOUS RESULT.
       117066  000000
22983  117070  000200                  4$:     200                             ;FPS BEFORE EXECUTION.
22984  117072  000204                          204                             ;FPS AFTER EXECUTION.
22985  117074  177777                          -1                              ;ANTICIPATED ERRONEOUS FPS.
22986                                  ;OPERAND=0      FL=1    FD=1
22987  117076  004737  117460          LLC3:   JSR     PC,@#LDCDSUB            ;GO EXECUTE THE INSTRUCTION.
22988  117102  000000  000000          1$:     .WORD   0,0                     ;FSRC OPERAND.
22989  117106  000000  000000  000000  2$:     .WORD   0,0,0,0                 ;EXPECTED RESULT.
       117114  000000
22990  117116  177777  177777  177777  3$:     .WORD   -1,-1,-1,-1                     ;ANTICIPATED ERRONEOUS RESULT.
       117124  177777
22991  117126  000211                  4$:     211                             ;FPS BEFORE EXECUTION.
22992  117130  000204                          204                             ;FPS AFTER EXECUTION.
22993  117132  177777                          -1                              ;ANTICIPATED ERRONEOUS FPS.
22994                                  ;OPERAND=40000  FL=0    FD=1
22995  117134  004737  117460          LLC4:   JSR     PC,@#LDCDSUB            ;GO EXECUTE THE INSTRUCTION.
22996  117140  040000  000000          1$:     .WORD   40000,0                 ;FSRC OPERAND.
22997  117144  043600  000000  000000  2$:     .WORD   43600,0,0,0                     ;EXPECTED RESULT.
       117152  000000
22998  117154  047600  000000  000000  3$:     .WORD   47600,0,0,0                     ;ANTICIPATED ERRONEOUS RESULT.
       117162  000000
22999  117164  000217                  4$:     217                             ;FPS BEFORE EXECUTION.
23000  117166  000200                          200                             ;FPS AFTER EXECUTION.
23001  117170  177777                          -1                              ;ANTICIPATED ERRONEOUS FPS.
23002                                  ;OPERAND=-40000 FL=0    FD=1
23003  117172  004737  117460          LLC5:   JSR     PC,@#LDCDSUB    ;GO EXECUTE THE INSTRUCTION.
23004  117176  140000  000000          1$:     .WORD   -40000,0                ;FSRC OPERAND.
23005  117202  143600  000000  000000  2$:     .WORD   143600,0,0,0            ;EXPECTED RESULT.
       117210  000000
23006  117212  043600  000000  000000  3$:     .WORD   43600,0,0,0                     ;ANTICIPATED ERRONEOUS RESULT.
       117220  000000
23007  117222  000200                  4$:     200                             ;FPS BEFORE EXECUTION.
23008  117224  000210                          210                             ;FPS AFTER EXECUTION.
23009  117226  177777                          -1                              ;ANTICIPATED ERRONEOUS FPS.
23010                                  ;OPERAND=40000,0        FL=1    FD=1
23011  117230  004737  117460          LLC6:   JSR     PC,@#LDCDSUB    ;GO EXECUTE THE INSTRUCTION.
23012  117234  040000  000000          1$:     .WORD   40000,0                 ;FSRC OPERAND.
23013  117240  047600  000000  000000  2$:     .WORD   47600,0,0,0                     ;EXPECTED RESULT.
       117246  000000
23014  117250  043600  000000  000000  3$:     .WORD   43600,0,0,0                     ;ANTICIPATED ERRONEOUS RESULT.
       117256  000000
23015  117260  000317                  4$:     317                             ;FPS BEFORE EXECUTION.
23016  117262  000300                          300                             ;FPS AFTER EXECUTION.
23017  117264  177777                          -1                              ;ANTICIPATED ERRONEOUS FPS.
23018                                  ;OPERAND=0,1    FL=1    FD=1
23019  117266  004737  117460          LLC7:   JSR     PC,@#LDCDSUB    ;GO EXECUTE THE INSTRUCTION.
23020  117272  000000  000001          1$:     .WORD   0,1                     ;FSRC OPERAND.
23021  117276  040200  000000  000000  2$:     .WORD   40200,0,0,0                     ;EXPECTED RESULT.
       117304  000000
23022  117306  034200  000000  000000  3$:     .WORD   34200,0,0,0                     ;ANTICIPATED ERRONEOUS RESULT.
       117314  000000
23023  117316  000300                  4$:     300                             ;FPS BEFORE EXECUTION.
```

```
23024  117320  000300                          300                     ;FPS AFTER EXECUTION.
23025  117322  177777                          -1                      ;ANTICIPATED ERRONEOUS FPS.
23026                                  ;OPERAND=77777,177777   FL=1    FD=1
23027  117324  004737  117460          LLC10:  JSR     PC,@#LDCDSUB    ;GO EXECUTE THE INSTRUCTION.
23028  117330  077777  177777          1$:     .WORD   77777,177777            ;FSRC OPERAND.
23029  117334  047777  177777  177000  2$:     .WORD   47777,177777,177000,0           ;EXPECTED RESULT.
       117342  000000
23030  117344  177777  177777  177777  3$:     .WORD   -1,-1,-1,-1             ;ANTICIPATED ERRONEOUS RESULT.
       117352  177777
23031  117354  000317                  4$:     317                     ;FPS BEFORE EXECUTION.
23032  117356  000300                          300                     ;FPS AFTER EXECUTION.
23033  117360  177777                          -1                      ;ANTICIPATED ERRONEOUS FPS.
23034                                  ;OPERAND=-PATTERN       FL=1    FD=1
23035
23036  117362  004767  000072          LLC11:  JSR     PC,LDCDSUB      ;GO EXECUTE THE INSTRUCTION.
23037  117366  177777  177526          1$:     .WORD   -1,-252         ;FSRC OPERAND.
23038  117372  142052  000000  000000  2$:     .WORD   142052,0,0,0            ;EXPECTED RESULT.
       117400  000000
23039  117402  136052  000000  000000  3$:     .WORD   136052,0,0,0            ;ANTICIPATED ERRONEOUS RESULT.
       117410  000000
23040  117412  000307                  4$:     307                     ;FPS BEFORE EXECUTION.
23041  117414  000310                          310                     ;FPS AFTER EXECUTION.
23042  117416  177777                          -1                      ;ANTICIPATED ERRONEOUS FPS.
23043                                  ;OPERAND=PATTERN        FL=1    FD=1    FT=1
23044  117420  004767  000034          LLC12:  JSR     PC,LDCDSUB      ;GO EXECUTE THE INSTRUCTION.
23045  117424  012345  067012          1$:     .WORD   12345,67012             ;FSRC OPERAND.
23046  117430  047247  025560  050000  2$:     .WORD   47247,025560,050000,0           ;EXPECTED RESULT.
       117436  000000
23047  117440  177777  177777  177777  3$:     .WORD   -1,-1,-1,-1             ;ANTICIPATED ERRONEOUS RESULT.
       117446  177777
23048  117450  000352                  4$:     352                     ;FPS BEFORE EXECUTION.
23049  117452  000340                          340                     ;FPS AFTER EXECUTION.
23050  117454  177777                          -1                      ;ANTICIPATED ERRONEOUS FPS.
23051  117456  000435                  6$:     BR      LLCDONE
23052
23053                                  ;THIS SUBROUTINE, LDCDSUB, IS USED TO SET UP THE OPERANDS, EXECUTE
23054                                  ;THE LDCID OR LDCLD INSTRUCTION AND CHECK THE RESULTS. A CALL
23055                                  ;TO IT IS MADE THUS:
23056                                  ;
23057                                  ;               JSR     PC,@#LDCDSUB
23058                                  ;               ACARG:  .WORD   X,X             ;AC OPERAND
23059                                  ;               RES:    .WORD   X,X,X,X         ;EXPECTED RESULT
23060                                  ;               ERRES:  .WORD   X,X,X,X         ;ERROR RESULT
23061                                  ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
23062                                  ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
23063                                  ;               ERFPS:  .WORD   X               ;ERROR FPS.
23064                                  ;               ERR1:   ERROR   X               ;DATA ERROR.
23065                                  ;                       BR      CONT
23066                                  ;               ERR2:   ERROR   X               ;FPS ERROR.
23067                                  ;               CONT:                           ;RETURN ADDRESS
23068                                  ;
23069                                  ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
23070                                  ;THE LDCID OR LDCLD INSTRUCTION IS EXECUTED.
23071                                  ;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
23072                                  ;COMPARED WITH FPSA IF THIS TOO IS CORRECT LDCDSUB RETURNS CONTROL
23073                                  ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD LDCDSUB
```

```
23074                                   ;COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN LDCDSUB WILL RETURN
23075                                   ;TO THE ERROR CALL AT ERR2, OTHERWISE LDCDSUB ITSELF
23076                                   ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
23077                                   ;LDCID OR LDCLD IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
23078                                   ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
23079                                   ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN LDCDSUB
23080                                   ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
23081                                   ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND LDCDSUB WILL
23082                                   ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.
23083
23084  117460  012601                   LDCDSUB:    MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
23085  117462  016100  000024                       MOV     24(R1),R0       ;SET THE FPS.
23086  117466  170100                              LDFPS   R0
23087  117470  010100                              MOV     R1,R0
23088  117472  177010                   1$:        LDCID   (R0),AC0        ;TEST INSTRUCTION, LDCID OR LDCLD.
23089
23090  117474  170204                              STFPS   R4              ;GET FPS.
23091  117476  012700  116766                       MOV     #LDCT,R0        ;GET THE RESULT.
23092  117502  012702  000200                       MOV     #200,R2
23093  117506  170102                              LDFPS   R2
23094  117510  174010                              STD     AC0,(R0)
23095
23096                                   ;SEE IF THE RESULT IS CORRECT.
23097  117512  012702  116766                       MOV     #LDCT,R2
23098  117516  010100                              MOV     R1,R0
23099  117520  062700  000004                       ADD     #4,R0
23100  117524  012703  000002                       MOV     #2,R3
23101  117530  022022                   2$:        CMP     (R0)+,(R2)+
23102  117532  001006                              BNE     10$             ;BR IF INCORRECT.
23103  117534  077303                              SOB     R3,2$
23104
23105  117536  026104  000026                       CMP     26(R1),R4               ;IS THE FPS CORRECT?
23106  117542  001002                              BNE     10$             ;BR IF INCORRECT.
23107  117544  000161  000032             3$:       JMP     32(R1)          ;RETURN.
23108  117550                           10$:
  (2)  117550  104000                              EMT                     ;
23109
23110  117552                           LLCDONE:
  (1)  117552  004767  004774                       JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                              ;SEE IF THE USER HAS EXPRESSED
  (1)                                              ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                              ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                              ;THE USER TYPED CONTROL G?).
23111
23120
23121                                   ;*****************************************************************************
  (2)                                   ;TEST 571       LDEXP TEST
  (3)                                   ;*****************************************************************************
  (2)  117556                      TS571:
23122
23123                                   ; NON-ZERO RES. VALID EXPON=210 (EXCESS 200)=10
23124  117536  004767  001136           MMC1:    JSR     PC,LDXSUB       ;GO EXECUTE THE INSTRUCTION.
23125  117562  012345  067012  034567   1$:      .WORD   12345,67012,34567,012345        ;AC0 OPERAND.
       117570  012345
23126  117572  000010                   2$:      .WORD   10              ;EXPONENT OPERAND.
23127  117574  042145  067012  034567   3$:      .WORD   42145,67012,34567,012345                ;EXPECTED RESULT.
```

```
                117602  012345
23128           117604  002145  067012  034567  4$:     .WORD   2145,67012,34567,012345         ;ANTICIPATED ERRONEOUS RESULT.
                117612  012345
23129           117614  047217                  5$:     47217                   ;FPS BEFORE EXECUTION.
23130           117616  047200                          47200                   ;FPS AFTER EXECUTION.
23131           117620  147200                          147200                  ;ANTICIPATED ERRONEOUS FPS.
23132           117622  177777                          -1              ;EXPECTED FEC.
23133                                           ;NON-ZERO       RES     NEG.
23134           117624  004737  120720          MMC2:   JSR     PC,@#LDXSUB     ;EXPON=377
23135           117630  123456  070123  045670 1$:     .WORD   123456,70123,45670,123456       ;ACO OPERAND.
                117636  123456
23136           117640  000177                  2$:     .WORD   177             ;EXPONENT OPERAND.
23137           117642  177656  070123  045670 3$:     .WORD   177656,70123,45670,123456               ;EXPECTED RESULT.
                117650  123456
23138           117652  137656  070123  045670 4$:     .WORD   137656,70123,45670,123456               ;ANTICIPATED ERRONEOUS RESULT.
                117660  123456
23139           117662  047207                  5$:     47207                   ;FPS BEFORE EXECUTION.
23140           117664  047210                          47210                   ;FPS AFTER EXECUTION.
23141           117666  147210                          147210                  ;ANTICIPATED ERRONEOUS FPS.
23142           117670  177777                          -1              ;EXPECTED FEC.
23143                                           ;NON-ZERO       RES,    EXP=256-(56)REAL
23144           117672  004737  120720          MMC3:   JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
23145           117676  073261  057645  043323 1$:     .WORD   73261,057645,43323,101760       ;ACO OPERAND.
                117704  101760
23146           117706  000056                  2$:     .WORD   56              ;EXPONENT OPERAND.
23147           117710  053461  057645  043323 3$:     .WORD   53461,057645,43323,101760               ;EXPECTED RESULT.
                117716  101760
23148           117720  177777  177777  177777 4$:     .WORD   -1,-1,-1,-1             ;ANTICIPATED ERRONEOUS RESULT.
                117726  177777
23149           117730  047200                  5$:     47200                   ;FPS BEFORE EXECUTION.
23150           117732  047200                          47200                   ;FPS AFTER EXECUTION.
23151           117734  147200                          147200                  ;ANTICIPATED ERRONEOUS FPS.
23152           117736  177777                          -1              ;EXPECTED FEC.
23153                                           ;EXP=27 (EXCESS 200)=-151 (OCT)
23154           117740  004737  120720          MMC4:   JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
23155           117744  012223  024252  062720 1$:     .WORD   12223,24252,62720,21222 ;ACO OPERAND.
                117752  021222
23156           117754  177627                  2$:     .WORD   -151            ;EXPONENT OPERAND.
23157           117756  005623  024252  062720 3$:     .WORD   5623,24252,62720,21222          ;EXPECTED RESULT.
                117764  021222
23158           117766  177777  177777  177777 4$:     .WORD   -1,-1,-1,-1             ;ANTICIPATED ERRONEOUS RESULT.
                117774  177777
23159           117776  047200                  5$:     47200                   ;FPS BEFORE EXECUTION.
23160           120000  047200                          47200                   ;FPS AFTER EXECUTION.
23161           120002  147200                          147200                  ;ANTICIPATED ERRONEOUS FPS.
23162           120004  177777                          -1              ;EXPECTED FEC.
23163                                           ;EXP=0 (EXCESS 200)=-200 (OCT), POSITIVE FRAC
23164                                           ; FIV=1
23165           120006  004737  120720          MMC5:   JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
23166           120012  030131  032334  035363 1$:     .WORD   30131,32334,35363,73031 ;ACO OPERAND.
                120020  073031
23167           120022  177600                  2$:     .WORD   -200            ;EXPONENT OPERAND.
23168           120024  000131  032334  035363 3$:     .WORD   00131,32334,35363,73031         ;EXPECTED RESULT.
                120032  073031
23169           120034  000000  000000  000000 4$:     .WORD   0,0,0,0         ;ANTICIPATED ERRONEOUS RESULT.
                120042  000000
```

```
23170  120044  042200                  5$:     42200                   ;FPS BEFORE EXECUTION.
23171  120046  142204                          142204                  ;FPS AFTER EXECUTION.
23172  120050  042202                          42202                   ;ANTICIPATED ERRONEOUS FPS.
23173  120052  000012                          12              ;EXPECTED FEC.
23174                                  ;EXP=0  (EXCESS 200)=-200 (OCT), NEG FRACT,FIU=1
23175  120054  004737  120720          MMC6:   JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
23176  120060  140414  024344  045464  1$:     .WORD   140414,24344,45464,74045        ;AC0 OPERAND.
       120066  074045
23177  120070  177600                  2$:     .WORD   -200                    ;EXPONENT OPERAND.
23178  120072  100014  024344  045464  3$:     .WORD   100014,24344,45464,74045        ;-0             ;EXPECTED RESULT.
       120100  074045
23179  120102  000000  000000  000000  4$:     .WORD   0,0,0,0                 ;ANTICIPATED ERRONEOUS RESULT.
       120110  000000
23180  120112  042200                  5$:     42200                   ;FPS BEFORE EXECUTION.
23181  120114  142214                          142214                  ;FPS AFTER EXECUTION.
23182  120116  042214                          42214                   ;ANTICIPATED ERRONEOUS FPS.
23183  120120  000012                          12              ;EXPECTED FEC.
23184                                  ;EXP=0 (EXCESS 200)=-200 (OCT),POS FRAC, FIU=0
23185
23186  120122  004737  120720          MMC7:   JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
23187  120126  051525  035455  005675  1$:     .WORD   51525,35455,5675,05152  ;AC0 OPERAND.
       120134  005152
23188  120136  177600                  2$:     .WORD   -200                    ;EXPONENT OPERAND.
23189  120140  000000  000000  000000  3$:     .WORD   0,0,0,0                 ;EXPECTED RESULT.
       120146  000000
23190  120150  000125  035455  005675  4$:     .WORD   00125,35455,5675,05152          ;ANTICIPATED ERRONEOUS RESULT.
       120156  005152
23191  120160  045200                          45200                   ;FPS BEFORE EXECUTION.
23192  120162  045204                          45204                   ;FPS AFTER EXECUTION.
23193  120164  145204                          145204                  ;ANTICIPATED ERRONEOUS FPS.
23194  120166  177777                          -1              ;EXPECTED FEC.
23195                                  ;EXP=-1405 (EXCESS 200)=-1605 (OCT), FIU=1
23196  120170  004737  120720          MMC10:  JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
23197  120174  061626  062636  046566  1$:     .WORD   61626,62636,46566,67606 ;AC0 OPERAND.
       120202  067606
23198  120204  176173                  2$:     .WORD   -1605                   ;EXPONENT OPERAND.
23199  120206  076626  062636  046566  3$:     .WORD   76626,62636,46566,67606         ;EXPECTED RESULT.
       120214  067606
23200  120216  000000  000000  000000  4$:     .WORD   0,0,0,0                 ;ANTICIPATED ERRONEOUS RESULT.
       120224  000000
23201  120226  042200                  5$:     42200                   ;FPS BEFORE EXECUTION.
23202  120230  142200                          142200                  ;FPS AFTER EXECUTION.
23203  120232  042204                          42204                   ;ANTICIPATED ERRONEOUS FPS.
23204  120234  000012                          12              ;EXPECTED FEC.
23205                                  ;EXP=-17416 (EXCESS 200)=-17616 (OCT), FIU=0
23206  120236  004737  120720          MMC11:  JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
23207  120242  071727  037475  076777  1$:     .WORD   71727,37475,76777,17273 ;AC0 OPERAND.
       120250  017273
23208  120252  160162                  2$:     .WORD   -17616                  ;EXPONENT OPERAND.
23209  120254  000000  000000  000000  3$:     .WORD   0,0,0,0                 ;EXPECTED RESULT.
       120262  000000
23210  120264  074527  037475  076777  4$:     .WORD   74527,37475,76777,17273         ;ANTICIPATED ERRONEOUS RESULT.
       120272  017273
23211  120274  045200                  5$:     45200                   ;FPS BEFORE EXECUTION.
23212  120276  045204                          45204                   ;FPS AFTER EXECUTION.
23213  120300  145200                          145200                  ;ANTICIPATED ERRONEOUS FPS.
```

```
120302  177777                          -1                  ;EXPECTED FEC.
                                ;EXP=-1601 (EXCESS 200)=-2001 (OCT), FIU=1
120304  004737  120720          MMC12:  JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
120310  001020  030405  006070  1$:     .WORD   01020,30405,06070,00102 ;AC0 OPERAND.
120316  000102
120320  175777                  2$:     .WORD   -2001           ;EXPONENT OPERAND.
120322  037620  030405  006070  3$:     .WORD   37620,30405,06070,00102         ;EXPECTED RESULT.
120330  000102
120332  000000  000000  000000  4$:     .WORD   0,0,0,0         ;ANTICIPATED ERRONEOUS RESULT.
120340  000000
120342  042200                  5$:     42200                   ;FPS BEFORE EXECUTION.
120344  142200                          142200                  ;FPS AFTER EXECUTION.
120346  042204                          42204                   ;ANTICIPATED ERRONEOUS FPS.
120350  000012                          12                  ;EXPECTED FEC.
                                ;EXP=1206 (EXCESS 200)=1006 (OCT) FIV =1
120352  004737  120720          MMC13:  JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
120356  012131  014151  016171  1$:     .WORD   12131,14151,16171,10111 ;AC0 OPERAND.
120364  010111
120366  001006                  2$:     .WORD   1006            ;EXPONENT OPERAND.
120370  041531  014151  016171  3$:     .WORD   41531,14151,16171,10111         ;EXPECTED RESULT.
120376  010111
120400  000000  000000  000000  4$:     .WORD   0,0,0,0         ;ANTICIPATED ERRONEOUS RESULT.
120406  000000
120410  041200                  5$:     41200                   ;FPS BEFORE EXECUTION.
120412  141202                          141202                  ;FPS AFTER EXECUTION.
120414  041204                          41204                   ;ANTICIPATED ERRONEOUS FPS.
120416  000010                          10                  ;EXPECTED FEC.
                                ;EXP=16315 (EXCESS 200)=16115 (OCT) FIV=0
120420  004737  120720          MMC14:  JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
120424  027262  025242  023222  1$:     .WORD   27262,25242,23222,21202 ;AC0 OPERAND.
120432  021202
120434  016115                  2$:     .WORD   16115           ;EXPONENT OPERAND.
120436  000000  000000  000000  3$:     .WORD   0,0,0,0         ;EXPECTED RESULT.
120444  000000
120446  063262  025242  023222  4$:     .WORD   63262,25242,23222,21202         ;ANTICIPATED ERRONEOUS RESULT.
120454  021202
120456  046200                  5$:     46200                   ;FPS BEFORE EXECUTION.
120460  046206                          46206                   ;FPS AFTER EXECUTION.
120462  146202                          146202                  ;ANTICIPATED ERRONEOUS FPS.
120464  177777                          -1                  ;EXPECTED FEC.
                                ;EXP=11011 (EXCESS 200)=10611 (OCT) FIV=1

120466  004737  120720          MMC15:  JSR     PC,@#LDXSUB     ;GO EXECUTE THE INSTRUCTION.
120472  030313  032333  034353  1$:     .WORD   30313,32333,34353,36373 ;AC0 OPERAND.
120500  036373
120502  010611                  2$:     .WORD   10611           ;EXPONENT OPERAND.
120504  002313  032333  034353  3$:     .WORD   2313,32333,34353,36373          ;EXPECTED RESULT.
120512  036373
120514  000000  000000  000000  4$:     .WORD   0,0,0,0         ;ANTICIPATED ERRONEOUS RESULT.
120522  000000
120524  041200                  5$:     41200                   ;FPS BEFORE EXECUTION.
120526  141202                          141202                  ;FPS AFTER EXECUTION.
120530  041204                          41204                   ;ANTICIPATED ERRONEOUS FPS.
120532  000010                          10                  ;EXPECTED FEC.
                                ;EXP=17123 (EXCESS 200)=16723 (OCT) FIV=0
```

```
23258  120534  004737  120720          MMC16:  JSR     PC,@@LDXSUB     ;GO EXECUTE THE INSTRUCTION.
23259  120540  040414  042434  044454  1$:     .WORD   40414,42434,44454,46474 ;AC0 OPERAND.
       120546  046474
23260  120550  016723                  2$:     .WORD   16723           ;EXPONENT OPERAND.
23261  120552  000000  000000  000000  3$:     .WORD   0,0,0,0         ;EXPECTED RESULT.
       120560  000000
23262  120562  024614  042434  044454  4$:     .WORD   24614,42434,44454,46474         ;ANTICIPATED ERRONEOUS RESULT.
       120570  046474
23263  120572  046200                  5$:     46200                   ;FPS BEFORE EXECUTION.
23264  120574  046206                          46206                   ;FPS AFTER EXECUTION.
23265  120576  146202                          146202                  ;ANTICIPATED ERRONEOUS FPS.
23266  120600  177777                          -1              ;EXPECTED FEC.
23267                                  ;EXP=   254 (OCT)=      454 (EXCESS 200)        FIV=1
23268
23269  120602  004737  120720          MMC17:  JSR     PC,@@LDXSUB     ;GO EXECUTE THE INSTRUCTION.
23270  120606  050515  052535  054555  1$:     .WORD   50515,52535,54555,56575 ;AC0 OPERAND.
       120614  056575
23271  120616  000254                  2$:     .WORD   254             ;EXPONENT OPERAND.
23272  120620  013115  052535  054555  3$:     .WORD   13115,52535,54555,56575         ;EXPECTED RESULT.
       120626  056575
23273  120630  000000  000000  000000  4$:     .WORD   0,0,0,0         ;ANTICIPATED ERRONEOUS RESULT.
       120636  000000
23274  120640  041200                  5$:     41200                   ;FPS BEFORE EXECUTION.
23275  120642  141202                          141202                  ;FPS AFTER EXECUTION.
23276  120644  041204                          41204                   ;ANTICIPATED ERRONEOUS FPS.
23277  120646  000010                          10              ;EXPECTED FEC.
23278                                  ;EXP=   313 (OCT)=      513(EXCESS 200) FIV=0
23279
23280  120650  004737  120720          MMC20:  JSR     PC,@@LDXSUB     ;GO EXECUTE THE INSTRUCTION.
23281  120654  060616  062636  064656  1$:     .WORD   60616,62636,64656,66676 ;AC0 OPERAND.
       120662  066676
23282  120664  000313                  2$:     .WORD   313             ;EXPONENT OPERAND.
23283  120666  000000  000000  000000  3$:     .WORD   0,0,0,0         ;EXPECTED RESULT.
       120674  000000
23284  120676  022616  062636  064656  4$:     .WORD   22616,62636,64656,66676         ;ANTICIPATED ERRONEOUS RESULT.
       120704  066676
23285  120706  046200                  5$:     46200                   ;FPS BEFORE EXECUTION.
23286  120710  046206                          46206                   ;FPS AFTER EXECUTION.
23287  120712  146202                          146202                  ;ANTICIPATED ERRONEOUS FPS.
23288  120714  177777                          -1              ;EXPECTED FEC.
23289  120716  000457                          BR      MMCDONE
23290
23291                          ;THIS SUBROUTINE, LDXSUB, IS USED TO SET UP THE OPERANDS, EXECUTE
23292                          ;THE LDEXP INSTRUCTION AND CHECK THE RESULTS. A CALL
23293                          ;TO IT IS MADE THUS:
23294                          ;
23295                          ;               JSR     PC,@@LDXSUB
23296                          ;               ACARG:  .WORD   X,X,X,X         ;AC OPERAND
23297                          ;               EXP:    .WORD   X               ;EXPONENT
23298                          ;               RES:    .WORD   X,X,X,X         ;EXPECTED RESULT
23299                          ;               ERRES:  .WORD   X,X,X,X         ;ERROR RESULT
23300                          ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
23301                          ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
23302                          ;               ERFPS:  .WORD   X               ;ERROR FPS.
23303                          ;               FEC:    .WORD   X               ;EXPECTED FEC
23304                          ;               ERR1:   ERROR   X               ;DATA ERROR.
```

```
23305                                   ;                       BR      CONT
23306                                   ;               ERR2:   ERROR   X               ;FPS ERROR.
23307                                   ;               CONT:                           ;RETURN ADDRESS
23308                                   ;
23309                                   ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
23310                                   ;THE LDEXP INSTRUCTION IS EXECUTED.
23311                                   ;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
23312                                   ;COMPARED WITH FPSA IF THIS TOO IS CORRECT LDXSUB RETURNS CONTROL
23313                                   ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD LDXSUB
23314                                   ;COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN LDXSUB WILL RETURN
23315                                   ;TO THE ERROR CALL AT ERR2, OTHERWISE LDXSUB ITSELF
23316                                   ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
23317                                   ;LDEXP IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
23318                                   ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
23319                                   ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN LDXSUB
23320                                   ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
23321                                   ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND LDXSUB WILL
23322                                   ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.
23323
23324  120720  012601                  LDXSUB: MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
23325  120722  012700  000200                  MOV     #200,R0         ;LOAD THE AC0 OPERAND.
23326  120726  170100                        LDFPS   R0
23327  120730  010100                        MOV     R1,R0
23328  120732  172410                        LDD     (R0),AC0
23329  120734  016100  000032                MOV     32(R1),R0       ;SET UP THE FPS.
23330  120740  170100                        LDFPS   R0
23331  120742  010100                        MOV     R1,R0
23332  120744  062700  000010                ADD     #10,R0
23333
23334  120750  176410                  1$:   LDEXP   (R0),AC0        ;TEST INSTRUCTION.
23335
23336  120752  170204                        STFPS   R4              ;GET THE FPS.
23337  120754  170305                        STST    R5              ;GET THE FEC.
23338  120756  012700  000200                MOV     #200,R0         ;GET THE RESULT.
23339  120762  170100                        LDFPS   R0
23340  120764  012700  121046                MOV     #LDXT,R0
23341  120770  174010                        STD     AC0,(R0)
23342  120772  012702  121046                MOV     #LDXT,R2        ;SEE IF THE RESULT WAS CORRECT.
23343  120776  010103                        MOV     R1,R3
23344  121000  062703  000012                ADD     #12,R3
23345  121004  012700  000004                MOV     #4,R0
23346  121010  022223                  2$:   CMP     (R2)+,(R3)+
23347  121012  001014                        BNE     10$             ;BRANCH IF NOT CORRECT.
23348  121014  077003                        SOB     R0,2$
23349  121016  020461  000034                CMP     R4,34(R1)       ;SEE IF THE FPS WAS CORRECT.
23350  121022  001010                        BNE     10$             ;BRANCH IF NOT CORRECT.
23351  121024  005761  000034                TST     34(R1)
23352  121030  100003                        BPL     3$
23353  121032  020561  000040                CMP     R5,40(R1)       ;SEE IF THE FEC WAS CORRECT.
23354  121036  001002                        BNE     10$             ;BRANCH IF NOT CORRECT.
23355
23356  121040  000161  000042          3$:   JMP     42(R1)          ;RETURN.
23357  121044                          10$:
  (2)  121044  104000                        EMT                     ;
23358
23359                                   ;DATA BUFFER:
```

```
23360  121046  000000  000000  000000  LDXT:   .WORD   0,0,0,0
       121054  000000
23361
23362  121056                          MMCDONE:
  (1)  121056  004767  003470                  JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
23363
23364
23371
23372                                   ;*************************************************************************************
  (2)                                   ;TEST 572       DESTINATION MODES, MODE 1 (FL=0), TEST
  (3)                                   ;*************************************************************************************
  (2)  121062                           TS572:
23373
23374
23375  121062  012700  121152                  MOV     #NNCTB0,R0      ;SET UP THE DATA BUFFER.
23376  121066  012701  000006                  MOV     #6,R1
23377  121072  012720  177777          1$:     MOV     #-1,(R0)+
23378  121076  077103                          SOB     R1,1$
23379  121100  012700  102345                  MOV     #102345,R0
23380  121104  012737  121166  000004          MOV     #NNC10,@#ERRVECT ;SET UP FOR TRAPS TO 4.
23381  121112  170100                          LDFPS   R0              ;SET UP FPS.
23382  121114  012700  121156                  MOV     #NNCTB1,R0
23383
23384  121120  170210                  NNC2:   STFPS   (R0)            ;TEST INSTRUCTION.
23385  121122  020027  121156                  CMP     R0,#NNCTB1      ;IS R0 CORRECT?
23386  121126  001017                          BNE     NNC10           ;BRANCH IF NOT CORRECT.
23387  121130  023727  121156  102345          CMP     @#NNCTB1,#102345   ;IS RESULT CORRECT?
23388  121136  001013                          BNE     NNC10           ;BRANCH IF NOT CORRECT.
23389  121140  023727  121160  177777          CMP     @#NNCTB1+2,#-1  ;IS THE RESULT CORRECT?
23390  121146  001007                          BNE     NNC10           ;BRANCH IF NOT CORRECT.
23391  121150  000407                          BR      NNCDONE
23392
23393                                   ;TEST DATA BUFFER:
23394  121152  177777  177777          NNCTB0: .WORD   -1,-1
23395  121156  177777  177777  177777  NNCTB1: .WORD   -1,-1,-1,-1
       121164  177777
23396  121166                          NNC10:
  (2)  121166  104000                          EMT                     ;
23397
23398  121170                          NNCDONE:
  (1)  121170  004767  003356                  JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
23399
23400
23401                                   ;*************************************************************************************
  (2)                                   ;TEST 573       DESTINATION MODES, MODE 2 (FL=0), TEST
  (3)                                   ;*************************************************************************************
  (2)  121174                           TS573:
23402
```

```
23403
23404  121174  012700  121264              MOV     #OOCTBO,RO      ;SET UP THE DATA BUFFER.
23405  121200  012701  000006              MOV     #6,R1
23406  121204  012720  177777      1$:     MOV     #-1,(RO)+
23407  121210  077103                      SOB     R1,1$
23408  121212  012700  105412              MOV     #105412,RO
23409  121216  012737  121300  000004      MOV     #OOC10,@#ERRVECT  ;SET UP FOR TRAPS TO VECTOR 4.
23410  121224  170100                      LDFPS   RO              ;SET UP FPS.
23411  121226  012700  121270              MOV     #OOCTB1,RO
23412
23413  121232  170220              OOC2:   STFPS   (RO)+           ;TEST INSTRUCTION.
23414  121234  020027  121272              CMP     RO,#OOCTB1+2    ;IS RO CORRECT?
23415  121240  001017                      BNE     OOC10           ;BRANCH IF NOT CORRECT.
23416  121242  023727  121270  105412      CMP     @#OOCTB1,#105412  ;IS THE RESULT CORRECT?
23417  121250  001013                      BNE     OOC10           ;BRANCH IF NOT CORRECT.
23418  121252  023727  121272  177777      CMP     @#OOCTB1+2,#-1  ;IS THE RESULT CORRECT?
23419  121260  001007                      BNE     OOC10           ;BRANCH IF NOT CORRECT.
23420  121262  000407                      BR      OOCDONE
23421
23422                          ;TEST DATA BUFFER:
23423  121264  177777  177777  OOCTBO: .WORD   -1,-1
23424  121270  177777  177777  177777  OOCTB1: .WORD   -1,-1,-1,-1
       121276  177777
23425  121300                  OOC10:
  (2)  121300  104000                      EMT                     ;
23426
23427  121302                  OOCDONE:
  (1)  121302  004767  003244              JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                              ;SEE IF THE USER HAS EXPRESSED
  (1)                                                              ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                              ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                              ;THE USER TYPED CONTROL G?).
23428
23429
23430
23431                          ;*****************************************************************************
  (2)                          ;TEST 574       DESTINATION MODES, MODE 4 (FL=0), TEST
  (3)                          ;*****************************************************************************
  (2)  121306                  TS574:
23432
23433  121306  012700  121376              MOV     #PPCTBO,RO      ;SET UP THE DATA BUFFER.
23434  121312  012701  000006              MOV     #6,R1
23435  121316  012720  177777      1$:     MOV     #-1,(RO)+
23436  121322  077103                      SOB     R1,1$
23437  121324  012700  105555              MOV     #105555,RO
23438  121330  012737  121412  000004      MOV     #PPC10,@#ERRVECT  ;SET UP FOR TRAPS TO VECTOR 4.
23439  121336  170100                      LDFPS   RO              ;SET UP FPS.
23440  121340  012700  121404              MOV     #PPCTB1+2,RO
23441
23442  121344  170240              PPC2:   STFPS   -(RO)           ;TEST INSTRUCTION.
23443  121346  020027  121402              CMP     RO,#PPCTB1      ;IS RO CORRECT?
23444  121352  001017                      BNE     PPC10           ;BRANCH IF NOT CORRECT.
23445  121354  023727  121402  105555      CMP     @#PPCTB1,#105555  ;IS THE RESULT CORRECT?
23446  121362  001013                      BNE     PPC10           ;BRANCH IF NOT CORRECT.
23447  121364  023727  121404  177777      CMP     @#PPCTB1+2,#-1  ;IS THE RESULT CORRECT?
23448  121372  001007                      BNE     PPC10           ;BRANCH IF NOT CORRECT.
```

```
23449  121374  000407                          BR      PPCDONE
23450
23451                                  ;TEST DATA BUFFER:
23452  121376  177777  177777          PPCTB0: .WORD   -1,-1
23453  121402  177777  177777  177777  PPCTB1: .WORD   -1,-1,-1,-1
       121410  177777
23454  121412                          PPC10:
  (2)  121412  104000                          EMT                     ;
23455  121414                          PPCDONE:
  (1)  121414  004767  003132                  JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
23456
23457
23458
23459                                  ;*************************************************************************************
  (2)                                  ;TEST 575       DESTINATION MODES, MODE 3 (FL=0), TEST
  (3)                                  ;*************************************************************************************
  (2)  121420                          TS575:
23460
23461  121420  012700  121514                  MOV     #QQCTB0,R0      ;SET UP THE DATA BUFFER.
23462  121424  012701  000010                  MOV     #10,R1
23463  121430  012720  177777          1$:     MOV     #-1,(R0)+
23464  121434  077103                          SOB     R1,1$
23465  121436  012700  106653                  MOV     #106653,R0
23466  121442  012737  121534  000004          MOV     #QQC10,@#ERRVECT  ;SET UP FOR TRAPS TO VECTOR 4.
23467  121450  170100                          LDFPS   R0              ;SET UP FPS.
23468  121452  012700  121530                  MOV     #QQCTB2,R0
23469  121456  012710  121520                  MOV     #QQCTB1,(R0)
23470
23471  121462  170230                  QQC2:   STFPS   @(R0)+          ;TEST INSTRUCTION.
23472  121464  020027  121532                  CMP     R0,#QQCTB2+2    ;IS R0 CORRECT?
23473  121470  001021                          BNE     QQC10           ;BRANCH IF NOT CORRECT.
23474  121472  023727  121520  106653          CMP     @#QQCTB1,#106653  ;IS THE RESULT CORRECT?
23475  121500  001015                          BNE     QQC10           ;BRANCH IF NOT CORRECT.
23476  121502  023727  121530  121520          CMP     @#QQCTB2,#QQCTB1  ;IS THE RESULT CORRECT?
23477  121510  001011                          BNE     QQC10           ;BRANCH IF NOT CORRECT.
23478  121512  000411                          BR      QQCDONE
23479
23480                                  ;TEST DATA BUFFER:
23481  121514  177777  177777          QQCTB0: .WORD   -1,-1
23482  121520  177777  177777  177777  QQCTB1: .WORD   -1,-1,-1,-1
       121526  177777
23483  121530  177777  177777          QQCTB2: .WORD   -1,-1
23484  121534                          QQC10:
  (2)  121534  104000                          EMT                     ;
23485  121536                          QQCDONE:
  (1)  121536  004767  003010                  JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
23486
23487
```

```
23488
23489                                   ;*****************************************************************************
  (2)                                   ;TEST 576       DESTINATION MODES, MODE 5 (FL=0), TEST
  (3)                                   ;*****************************************************************************
  (2)  121542                           TS576:
23490
23491
23492  121542  012700  121640                 MOV     #RRCTB0,R0      ;SET UP THE DATA BUFFER.
23493  121546  012701  000006                 MOV     #6,R1
23494  121552  012720  177777           1$:   MOV     #-1,(R0)+
23495  121556  077103                         SOB     R1,1$
23496  121560  012700  004301                 MOV     #004301,R0
23497  121564  012737  121660  000004         MOV     #RRC10,@#ERRVECT  ;SET UP FOR TRAPS TO VECTOR 4.
23498  121572  170100                         LDFPS   R0              ;SET UP FPS.
23499  121574  012700  121656                 MOV     #RRCTB2+2,R0
23500  121600  012760  121644  177776         MOV     #RRCTB1,-2(R0)
23501
23502  121606  170250                   RRC2: STFPS   @-(R0)          ;TEST INSTRUCTION.
23503  121610  020027  121654                 CMP     R0,#RRCTB2      ;IS R0 CORRECT?
23504  121614  001021                         BNE     RRC10           ;BRANCH IF NOT CORRECT.
23505  121616  023727  121644  004301         CMP     @#RRCTB1,#004301  ;IS THE RESULT CORRECT?
23506  121624  001015                         BNE     RRC10           ;BRANCH IF NOT CORRECT.
23507  121626  023727  121654  121644         CMP     @#RRCTB2,#RRCTB1  ;IS THE RESULT CORRECT?
23508  121634  001011                         BNE     RRC10           ;BRANCH IF NOT CORRECT.
23509  121636  000411                         BR      RRCDONE
23510
23511                                   ;TEST DATA BUFFER:
23512  121640  177777  177777           RRCTB0: .WORD   -1,-1
23513  121644  177777  177777  177777   RRCTB1: .WORD   -1,-1,-1,-1
       121652  177777
23514  121654  177777  177777           RRCTB2: .WORD   -1,-1
23515  121660                           RRC10:
  (2)  121660  104000                         EMT                     ;
23516  121662                           RRCDONE:
  (1)  121662  004767  002664                 JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
23517
23518
23519                                   ;*****************************************************************************
  (2)                                   ;TEST 577       DESTINATION MODES, MODE 6 (FL=0), TEST
  (3)                                   ;*****************************************************************************
  (2)  121666                           TS577:
23520
23521
23522  121666  012700  121770                 MOV     #SSCTB0,R0      ;SET UP THE DATA BUFFER.
23523  121672  012701  000006                 MOV     #6,R1
23524  121676  012720  177777           1$:   MOV     #-1,(R0)+
23525  121702  077103                         SOB     R1,1$
23526  121704  012700  102514                 MOV     #102514,R0
23527  121710  012737  122004  000004         MOV     #SSC10,@#ERRVECT  ;SET UP FOR TRAPS TO VECTOR 4.
23528  121716  170100                         LDFPS   R0              ;SET UP FPS.
23529  121720  005001                         CLR     R1
23530  121722  012700  114573                 MOV     #SSCTB1-5201,R0
```

```
23531
23532  121726  170260  005201          SSC2:   STFPS   5201(R0)                 ;TEST INSTRUCTION.
23533  121732  020127  000000                  CMP     R1,#0           ;WAS PC CORRECT AFTER EXECUTION?
23534  121736  001022                          BNE     SSC10           ;BRANCH IF NOT CORRECT.
23535  121740  020027  114573                  CMP     R0,#SSCTB1-5201 ;IS R0 CORRECT?
23536  121744  001017                          BNE     SSC10           ;BRANCH IF NOT CORRECT.
23537  121746  023727  121774  102514          CMP     @#SSCTB1,#102514  ;IS THE RESULT CORRECT?
23538  121754  001013                          BNE     SSC10           ;BRANCH IF NOT CORRECT.
23539  121756  023727  121776  177777          CMP     @#SSCTB1+2,#-1  ;IS THE RESULT CORRECT?
23540  121764  001007                          BNE     SSC10           ;BRANCH IF NOT CORRECT.
23541  121766  000407                          BR      SSCDONE
23542
23543                                  ;TEST DATA BUFFER:
23544  121770  177777  177777          SSCTB0: .WORD   -1,-1
23545  121774  177777  177777  177777  SSCTB1: .WORD   -1,-1,-1,-1
       122002  177777
23546  122004                          SSC10:
  (2)  122004  104000                          EMT                     ;
23547  122006                          SSCDONE:
  (1)  122006  004767  002540                  JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
```

```
                                        ;*******************************************************************************
                                        ;TEST 600       DESTINATION MODES, MODE 7 (FL=0), TEST
                                        ;*******************************************************************************
  122012                                TS600:

  122012  012700  122122                        MOV     #TTCTB0,R0          ;SET UP THE DATA BUFFER.
  122016  012701  000010                        MOV     #10,R1
  122022  012720  177777                1$:     MOV     #-1,(R0)+
  122026  077103                                SOB     R1,1$
  122030  012700  103747                        MOV     #103747,R0
  122034  012737  122142  000004                MOV     #TTC10,@#ERRVECT  ;SET UP FOR TRAPS TO VECTOR 4.
  122042  170100                                LDFPS   R0                  ;SET UP FPS.
  122044  005001                                CLR     R1
  122046  012700  114735                        MOV     #TTCTB2-5201,R0
  122052  012760  122126  005201                MOV     #TTCTB1,5201(R0)

  122060  170270  005201                TTC2:   STFPS   @5201(R0)                   ;TEST INSTRUCTION.
  122064  022701  000000                        CMP     #0,R1               ;WAS PC CORRECT AFTER EXECUTION?
  122070  001024                                BNE     TTC10               ;BRANCH IF NOT CORRECT.
  122072  020027  114735                        CMP     R0,#TTCTB2-5201     ;IS R0 CORRECT?
  122076  001021                                BNE     TTC10               ;BRANCH IF NOT CORRECT.
  122100  023727  122126  103747                CMP     @#TTCTB1,#103747  ;IS THE RESULT CORRECT?
  122106  001015                                BNE     TTC10               ;BRANCH IF NOT CORRECT.
  122110  023727  122130  177777                CMP     @#TTCTB1+2,#-1      ;IS THE RESULT CORRECT?
  122116  001011                                BNE     TTC10               ;BRANCH IF NOT CORRECT.
  122120  000411                                BR      TTCDONE

                                        ;TEST DATA BUFFER:
  122122  177777  177777                TTCTB0: .WORD   -1,-1
  122126  177777  177777  177777        TTCTB1: .WORD   -1,-1,-1,-1
  122134  177777
  122136  177777  177777                TTCTB2: .WORD   -1,-1
  122142                                TTC10:
  122142  104000                                EMT                         ;
  122144                                TTCDONE:
  122144  004767  002402                        JSR     PC,.RSET            ;GO INITIALIZE THE FPS AND STACK; AND
                                                                            ;SEE IF THE USER HAS EXPRESSED
                                                                            ;THE DESIRE TO CHANGE THE SOFTWARE
                                                                            ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
                                                                            ;THE USER TYPED CONTROL G?).

                                        ;*******************************************************************************
                                        ;TEST 601       DESTINATION MODES, MODE 2 (FL=1), TEST
                                        ;*******************************************************************************
  122150                                TS601:
  122150  012700  000300                        MOV     #300,R0             ;SET UP FPS.
  122154  170100                                LDFPS   R0
  122156  012700  122202                        MOV     #UUCTP1,R0          ;SET UP THE AC0 OPERAND.
  122162  172410                                LDD     (R0),AC0
  122164  012700  122214                        MOV     #UUCBF0,R0

  122170  175420                        UUC2:   STCDL   AC0,(R0)+           ;TEST INSTRUCTION.

  122172  020027  122220                        CMP     R0,#UUCBF0+4        ;IS R0 CORRECT?
```

```
23598  122176  001411                          BEQ     UUCDONE
  (2)  122200  104000                          EMT                     ;
23599                                 ;TEST DATA BUFFER:
23600  122202  000000  000000  000000  UUCTP1: .WORD   0,0,0,0
       122210  000000
                                              -1
23601  122212  177777
23602  122214  177777  177777  177777  UUCBF0: .WORD   -1,-1,-1
23603
23604  122222                          UUCDONE:
  (1)  122222  004767  002324                  JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
23605
23612                                 ;*****************************************************************************
  (2)                                 ;TEST 602       DESTINATION MODES, MODE 4 (FL=1), TEST
  (3)                                 ;*****************************************************************************
  (2)  122226                         TS602:
23613
23614  122226  012700  000300                  MOV     #300,R0         ;SET UP FPS.
23615  122232  170100                          LDFPS   R0
23616  122234  012700  122260                  MOV     #VVCTP1,R0      ;SET UP THE AC0 OPERAND.
23617  122240  172410                          LDD     (R0),AC0
23618  122242  012700  122276                  MOV     #VVCBF0+4,R0
23619
23620  122246  175440                  VVC2:   STCDL   AC0,-(R0)               ;TEST INSTRUCTION.
23621
23622  122250  020027  122272                  CMP     R0,#VVCBF0      ;IS R0 CORRECT?
23623  122254  001411                          BEQ     VVCDONE
  (2)  122256  104000                          EMT                     ;
23624                                 ;TEST DATA BUFFER:
23625  122260  000000  000000  000000  VVCTP1: .WORD   0,0,0,0
       122266  000000
                                              -1
23626  122270  177777
23627  122272  177777  177777  177777  VVCBF0: .WORD   -1,-1,-1
23628
23629  122300                          VVCDONE:
  (1)  122300  004767  002246                  JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
23630
23640                                 ;*****************************************************************************
  (2)                                 ;TEST 603       STCDI AND STCDL TEST
  (3)                                 ;*****************************************************************************
  (2)  122304                         TS603:
23641
23642                                 ;FIRST TEST STC WITH EXP=100 (EXCESS 200)
23643  122304  004737  123242          WWC1:   JSR     PC,@#STCSUB     ;GO EXECUTE THE INSTRUCTION.
23644  122310  020000  000000  000000  1$:     .WORD   20000,0,0,0     ;AC0 OPERAND.
       122316  000000
23645  122320  000000  000000          2$:     .WORD   0,0             ;EXPECTED RESULT.
23646  122324  177777  177777          3$:     .WORD   -1,-1           ;ERROR RES.
23647  122330  040300                  4$:     40300                   ;FPS BEFORE EXECUTION.
```

```
23648  122332  040304                          40304                   ;FPS AFTER EXECUTION.
23649  122334  140304                          140304                  ;ANTICIPATED ERRONEOUS FPS.
23650  122336  177777                          -1              ;REPORT RESULT INCORRECT.
23651                                  ;EXP=0 (OCT)    FL=1    FIC=0
23652  122340  004737  123242          WWC2:   JSR     PC,@#STCSUB     ;GO EXECUTE THE INSTRUCTION.
23653  122344  040000  000000  000000  1$:     .WORD   40000,0,0,0     ;AC     ;AC0 OPERAND.
       122352  000000
23654  122354  000000  000000          2$:     .WORD   0,0             ;EXPECTED RESULT.
23655  122360  177777  177777          3$:     .WORD   -1,-1           ;ANTICIPATED ERRONEOUS RESULT.
23656  122364  040313                  4$:     40313                   ;FPS BEFORE EXECUTION.
23657  122366  040304                          40304                   ;FPS AFTER EXECUTION.
23658  122370  140304                          140304                  ;ANTICIPATED ERRONEOUS FPS.
23659  122372  177777                          -1              ;EXPECTED FEC.
23660                                  ;EXP=37 (OCT)   FL=1    FIC=1
23661  122374  004737  123242          WWC3:   JSR     PC,@#STCSUB     ;GO EXECUTE THE INSTRUCTION.
23662  122400  047667  075757  157737  1$:     .WORD   47667,75757,157737,167773       ;AC0 OPERAND.
       122406  167773
23663  122410  055675  173757          2$:     .WORD   55675,173757            ;EXPECTED RESULT.
23664  122414  122102  004021          3$:     .WORD   122102,004021           ;ANTICIPATED ERRONEOUS RESULT.
23665  122420  040717                  4$:     40717                   ;FPS BEFORE EXECUTION.
23666  122422  040700                          40700                   ;FPS AFTER EXECUTION.
23667  122424  140705                          140705                  ;ANTICIPATED ERRONEOUS FPS.
23668  122426  177777                          -1              ;EXPECTED FEC.
23669                                  ;EXP=40 (OCT)   FL=1    FIC=1
23670  122430  004737  123242          WWC4:   JSR     PC,@#STCSUB     ;GO EXECUTE THE INSTRUCTION.
23671  122434  050000  000000  000000  1$:     .WORD   50000,0,0,0     ;AC0 OPERAND.
       122442  000000
23672  122444  000000  000000          2$:     .WORD   0,0             ;EXPECTED RESULT.
23673  122450  177777  177777          3$:     .WORD   -1,-1           ;ANTICIPATED ERRONEOUS RESULT.
23674  122454  040700                  4$:     40700                   ;FPS BEFORE EXECUTION.
23675  122456  140705                          140705                  ;FPS AFTER EXECUTION.
23676  122460  040705                          40705                   ;ANTICIPATED ERRONEOUS FPS.
23677  122462  000006                          6               ;EXPECTED FEC.
23678
23679                                  ;EXP=40 (OCT)    FL=1   FIC=0
23680  122464  004737  123242          WWC5:   JSR     PC,@#STCSUB     ;GO EXECUTE THE INSTRUCTION.
23681  122470  050000  000000  000000  1$:     .WORD   50000,0,0,0     ;AC0 OPERAND.
       122476  000000
23682  122500  000000  000000          2$:     .WORD   0,0             ;EXPECTED RESULT.
23683  122504  177777  177777          3$:     .WORD   -1,-1           ;ANTICIPATED ERRONEOUS RESULT.
23684  122510  040312                  4$:     40312                   ;FPS BEFORE EXECUTION.
23685  122512  040305                          40305                   ;FPS AFTER EXECUTION.
23686  122514  140305                          140305                  ;ANTICIPATED ERRONEOUS FPS.
23687  122516  177777                          -1              ;EXPECTED FEC.
23688                                  ;EXP=30 (OCT)   FL=1    FIC=1
23689  122520  004737  123242          WWC6:   JSR     PC,@#STCSUB     ;GO EXECUTE THE INSTRUCTION.
23690  122524  046000  000001  000000  1$:     .WORD   46000,1,0,0     ;AC0 OPERAND.
       122532  000000
23691  122534  000200  000001          2$:     .WORD   200,1           ;EXPECTED RESULT.
23692  122540  177777  177777          3$:     .WORD   -1,-1           ;ANTICIPATED ERRONEOUS RESULT.
23693  122544  040700                  4$:     40700                   ;FPS BEFORE EXECUTION.
23694  122546  040700                          40700                   ;FPS AFTER EXECUTION.
23695  122550  177777                          -1                      ;ANTICIPATED ERRONEOUS FPS.
23696  122552  177777                          -1              ;EXPECTED FEC.
23697                                  ;EXP=27 (OCT)   FL=1    FIC=1
23698  122554  004737  123242          WWC7:   JSR     PC,@#STCSUB     ;GO EXECUTE THE INSTRUCTION.
```

```
23699  122560  045600  000001  000000  1$:     .WORD   45600,1,0,0      ;ACO OPERAND.
       122566  000000
23700  122570  000100  000000          2$:     .WORD   100,0             ;EXPECTED RESULT.
23701  122574  177777  177777          3$:     .WORD   -1,-1             ;ANTICIPATED ERRONEOUS RESULT.
23702  122600  040707                  4$:     40707                     ;FPS BEFORE EXECUTION.
23703  122602  040700                          40700                     ;FPS AFTER EXECUTION.
23704  122604  177777                          -1                        ;ANTICIPATED ERRONEOUS FPS.
23705  122606  177777                          -1              ;EXPECTED FEC.
23706                                  ;EXP=17 (OCT)   FL=0    FIC=1
23707  122610  004737  123242          WWC10:  JSR     PC,@#STCSUB       ;GO EXECUTE THE INSTRUCTION.
23708  122614  043600  000000  000000  1$:     .WORD   43600,0,0,0       ;ACO OPERAND.
       122622  000000
23709  122624  040000  177777          2$:     .WORD   40000,-1                  ;EXPECTED RESULT.
23710  122630  000000  177777          3$:     .WORD   0,-1              ;ANTICIPATED ERRONEOUS RESULT.
23711  122634  040600                  4$:     40600                     ;FPS BEFORE EXECUTION.
23712  122636  040600                          40600                     ;FPS AFTER EXECUTION.
23713  122640  140604                          140604                    ;ANTICIPATED ERRONEOUS FPS.
23714  122642  177777                          -1              ;EXPECTED FEC.
23715
23716                                  ;EXP=20 (OCT)   FL=0    FIC=1
23717  122644  004737  123242          WWC11:  JSR     PC,@#STCSUB       ;GO EXECUTE THE INSTRUCTION.
23718  122650  044000  000000  000000  1$:     .WORD   44000,0,0,0       ;ACO OPERAND.
       122656  000000
23719  122660  000000  177777          2$:     .WORD   0,-1              ;EXPECTED RESULT.
23720  122664  177777  177777          3$:     .WORD   -1,-1             ;ANTICIPATED ERRONEOUS RESULT.
23721  122670  040600                  4$:     40600                     ;FPS BEFORE EXECUTION.
23722  122672  140605                          140605                    ;FPS AFTER EXECUTION.
23723  122674  040600                          40600                     ;ANTICIPATED ERRONEOUS FPS.
23724  122676  000006                          6               ;EXPECTED FEC.
23725                                  ;EXP=10 (OCT), AC NEGATIVE, FL=0, FIC=1
23726  122700  004737  123242          WWC12:  JSR     PC,@#STCSUB       ;GO EXECUTE THE INSTRUCTION.
23727  122704  142000  000000  000000  1$:     .WORD   142000,0,0,0      ;ACO OPERAND.
       122712  000000
23728  122714  177600  177777          2$:     .WORD   177600,-1                 ;EXPECTED RESULT.
23729  122720  000200  000000          3$:     .WORD   200,0             ;ANTICIPATED ERRONEOUS RESULT.
23730  122724  040600                  4$:     40600                     ;FPS BEFORE EXECUTION.
23731  122726  040610                          40610                     ;FPS AFTER EXECUTION.
23732  122730  040600                          40600                     ;ANTICIPATED ERRONEOUS FPS.
23733  122732  177777                          -1              ;EXPECTED FEC.
23734                                  ;EXP=37 (OCT), FL=1, FIC=1, AC NEG.
23735  122734  004737  123242          WWC13:  JSR     PC,@#STCSUB       ;GO EXECUTE THE INSTRUCTION.
23736  122740  147600  000000  000000  1$:     .WORD   147600,0,0,0      ;ACO OPERAND.
       122746  000000
23737  122750  140000  000000          2$:     .WORD   140000,0                  ;EXPECTED RESULT.
23738  122754  137777  000000          3$:     .WORD   137777,0                  ;ANTICIPATED ERRONEOUS RESULT.
23739  122760  040700                  4$:     40700                     ;FPS BEFORE EXECUTION.
23740  122762  040710                          40710                     ;FPS AFTER EXECUTION.
23741  122764  177777                          -1                        ;ANTICIPATED ERRONEOUS FPS.
23742  122766  177777                          -1              ;EXPECTED FEC.
23743                                  ;EXP=37  (OCT),  FL=1,  FIC=1,  AC  NEG
23744  122770  004737  123242          WWC14:  JSR     PC,@#STCSUB       ;GO EXECUTE THE INSTRUCTION.
23745  122774  147600  000000  001000  1$:     .WORD   147600,0,1000,0   ;ACO OPERAND.
       123002  000000
23746  123004  137777  177777          2$:     .WORD   137777,177777             ;EXPECTED RESULT.
23747  123010  140000  177777          3$:     .WORD   140000,177777             ;ANTICIPATED ERRONEOUS RESULT.
23748  123014  040707                  4$:     40707                     ;FPS BEFORE EXECUTION.
```

```
23749  123016  040710                          40710                   ;FPS AFTER EXECUTION.
23750  123020  177777                          -1                      ;ANTICIPATED ERRONEOUS FPS.
23751  123022  177777                          -1              ;EXPECTED FEC.
23752                                  ;EXP=41 (OCT),  AC NEG, FL=1, FIC=1
23753  123024  004737  123242                  WWC15:  JSR     PC,@#STCSUB     ;GO EXECUTE THE INSTRUCTION.
23754  123030  150200  000000  000000          1$:     .WORD   150200,0,0,0    ;ACO OPERAND.
       123036  000000
23755  123040  000000  000000                  2$:     .WORD   0,0             ;EXPECTED RESULT.
23756  123044  177777  177777                  3$:     .WORD   -1,-1           ;ANTICIPATED ERRONEOUS RESULT.
23757  123050  040700                          4$:     40700                   ;FPS BEFORE EXECUTION.
23758  123052  140705                          140705                  ;FPS AFTER EXECUTION.
23759  123054  177777                          -1                      ;ANTICIPATED ERRONEOUS FPS.
23760  123056  000006                          6               ;EXPECTED FEC.
23761                                  ;EXP=40 (OCT),  AC NEG,  FL=1,  FIC=1
23762  123060  004737  123242                  WWC16:  JSR     PC,@#STCSUB     ;GO EXECUTE THE INSTRUCTION.
23763  123064  150000  000001  000000          1$:     .WORD   150000,1,0,0    ;ACO OPERAND.
       123072  000000
23764  123074  000000  000000                  2$:     .WORD   0,0             ;EXPECTED RESULT.
23765  123100  100000  177600                  3$:     .WORD   100000,-200             ;ANTICIPATED ERRONEOUS RESULT.
23766  123104  040700                          4$:     40700                   ;FPS BEFORE EXECUTION.
23767  123106  140705                          140705                  ;FPS AFTER EXECUTION.
23768  123110  040700                          40700                   ;ANTICIPATED ERRONEOUS FPS.
23769  123112  000006                          6               ;EXPECTED FEC.
23770                                  ;EXP=40, AC NEGATIVE, FL=1, FIC=1
23771  123114  004737  123242                  WWC17:  JSR     PC,@#STCSUB     ;GO EXECUTE THE INSTRUCTION.
23772  123120  150001  000000  000000          1$:     .WORD   150001,0,0,0    ;ACO OPERAND.
       123126  000000
23773  123130  000000  000000                  2$:     .WORD   0,0             ;EXPECTED RESULT.
23774  123134  077400  000000                  3$:     .WORD   77400,0         ;ANTICIPATED ERRONEOUS RESULT.
23775  123140  040700                          4$:     40700                   ;FPS BEFORE EXECUTION.
23776  123142  140705                          140705                  ;FPS AFTER EXECUTION.
23777  123144  177777                          -1                      ;ANTICIPATED ERRONEOUS FPS.
23778  123146  000006                          6               ;EXPECTED FEC.
23779                                  ;EXP 40    (OCT), AC MOST NEG LONG INT, FL=1
23780                                  ;FIC=1
23781  123150  004737  123242                  WWC20:  JSR     PC,@#STCSUB     ;GO EXECUTE THE INSTRUCTION.
23782  123154  150000  000000  000000          1$:     .WORD   150000,0,0,0    ;ACO OPERAND.
       123162  000000
23783  123164  100000  000000                  2$:     .WORD   100000,0                ;EXPECTED RESULT.
23784  123170  000000  000000                  3$:     .WORD   0,0             ;ANTICIPATED ERRONEOUS RESULT.
23785  123174  040700                          4$:     40700                           ;FPS BEFORE EXECUTION.
23786  123176  040710                          40710                   ;FPS AFTER EXECUTION.
23787  123200  140705                          140705                  ;ANTICIPATED ERRONEOUS FPS.
23788  123202  177777                          -1              ;EXPECTED FEC.
23789                                  ;EXP=20, AC = MOST NEG INTEGER, FL=0, FIC=1
23790
23791  123204  004737  123242                  WWC21:  JSR     PC,@#STCSUB     ;GO EXECUTE THE INSTRUCTION.
23792  123210  144000  000001  000000          1$:     .WORD   144000,1,0,0    ;ACO OPERAND.
       123216  000000
23793  123220  100000  177777                  2$:     .WORD   100000,-1               ;EXPECTED RESULT.
23794  123224  100000  177400                  3$:     .WORD   100000,177400           ;ANTICIPATED ERRONEOUS RESULT.
23795  123230  040600                          4$:     40600                   ;FPS BEFORE EXECUTION.
23796  123232  040610                          40610                   ;FPS AFTER EXECUTION.
23797  123234  140605                          140605                  ;ANTICIPATED ERRONEOUS FPS.
23798  123236  177777                          -1              ;EXPECTED FEC.
23799  123240  000457                          6$:     BR      WWCDONE
```

```
23800
23801                           ;THIS SUBROUTINE, STCSUB, IS USED TO SET UP THE OPERANDS, EXECUTE
23802                           ;THE STCDI OR STCDL INSTRUCTION AND CHECK THE RESULTS. A CALL
23803                           ;TO IT IS MADE THUS:
23804                           ;
23805                           ;       JSR     PC,@#STCSUB
23806                           ;       ACARG:  .WORD   X,X,X,X         ;AC OPERAND
23807                           ;       RES:    .WORD   X,X             ;EXPECTED RESULT
23808                           ;       ERRES:  .WORD   X,X             ;ERROR RESULT
23809                           ;       FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
23810                           ;       FPSA:   .WORD   X               ;FPS AFTER EXECUTION
23811                           ;       ERFPS:  .WORD   X               ;ERROR FPS.
23812                           ;       FEC:    .WORD   X               ;EXPECTED FEC
23813                           ;       ERR1:   ERROR   X               ;DATA ERROR.
23814                           ;               BR      CONT
23815                           ;       ERR2:   ERROR   X               ;FPS ERROR.
23816                           ;       CONT:                           ;RETURN ADDRESS
23817                           ;
23818                           ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
23819                           ;THE STCDI OR STCDL INSTRUCTION IS EXECUTED.
23820                           ;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
23821                           ;COMPARED WITH FPSA IF THIS TOO IS CORRECT STCSUB RETURNS CONTROL
23822                           ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD STCSUB
23823                           ;COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN STCSUB WILL RETURN
23824                           ;TO THE ERROR CALL AT ERR2, OTHERWISE STCSUB ITSELF
23825                           ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
23826                           ;STCDI OR STCDL IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
23827                           ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
23828                           ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN STCSUB
23829                           ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
23830                           ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND STCSUB WILL
23831                           ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.
23832
23833  123242  012601          STCSUB: MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
23834  123244  012700  000200          MOV     #200,R0         ;SET UP THE AC0 OPERAND.
23835  123250  170100                  LDFPS   R0
23836  123252  010100                  MOV     R1,R0
23837  123254  172410                  LDD     (R0),AC0
23838  123256  012702  123370          MOV     #STCIBF,R2      ;INITIALIZE THE OUT PUT BUFFER.
23839  123262  012700  000004          MOV     #4,R0
23840  123266  012722  177777  1$:     MOV     #-1,(R2)+
23841  123272  077003                  SOB     R0,1$
23842  123274  016100  000020          MOV     20(R1),R0       ;SET THE FPS.
23843  123300  170100                  LDFPS   R0
23844  123302  012700  123370          MOV     #STCIBF,R0
23845  123306  175410          2$:     STCDL   AC0,(R0)        ;TEST INSTRUCTION.
23846
23847  123310  170204                  STFPS   R4              ;GET THE FPS.
23848  123312  170305                  STST    R5              ;GET THE FEC.
23849  123314  010102                  MOV     R1,R2
23850  123316  062702  000010          ADD     #10,R2
23851  123322  012700  123370          MOV     #STCIBF,R0      ;SEE IF THE RESULT IS CORRECT.
23852  123326  012703  000002          MOV     #2,R3
23853  123332  022022          3$:     CMP     (R0)+,(R2)+
23854  123334  001014                  BNE     10$
23855  123336  077303                  SOB     R3,3$
```

```
23856  123340  016102  000022                  MOV     22(R1),R2
23857  123344  020204                          CMP     R2,R4           ;SEE IF THE FPS IS CORRECT.
23858  123346  001007                          BNE     10$             ;BRANCH IF INCORRECT.
23859  123350  005702                          TST     R2
23860  123352  100003                          BPL     4$
23861  123354  026105  000026                  CMP     26(R1),R5       ;SEE IF THE FEC IS CORRECT.
23862  123360  001002                          BNE     10$             ;BRANCH IF INCORRECT.
23863
23864  123362  000161  000030          4$:     JMP     30(R1)          ;RETURN.
23865  123366                          10$:
  (2)  123366  104000                          EMT                     ;
23866
23867                                  ;DATA BUFFER:
23868  123370  177777  177777  177777  STCIBF: .WORD   -1,-1,-1,-1
       123376  177777
23869
23870  123400                          WWCDONE:
  (1)  123400  004767  001146                  JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
23871
23872
23880                                  ;*************************************************************************************
  (2)                                  ;TEST 604       STCFL AND STCFI TEST
  (3)                                  ;*************************************************************************************
  (2)  123404                          TS604:
23881
23882
23883                                  ;EXPONENT=37, FL=1
23884  123404  004737  123242                  JSR     PC,@#STCSUB     ;GO EXECUTE THE INSTRUCTION.
23885  123410  047777  177777  177777  1$:     .WORD   47777,-1,-1,-1  ;ACO OPERAND.
       123416  177777
23886  123420  077777  177600          2$:     .WORD   77777,177600            ;EXPECTED RESULT.
23887  123424  077777  177777          3$:     .WORD   77777,177777            ;ANTICIPATED ERRONEOUS RESULT.
23888  123430  040100                  4$:     40100                   ;FPS BEFORE EXECUTION.
23889  123432  040100                          40100                   ;FPS AFTER EXECUTION.
23890  123434  177777                          -1                      ;ANTICIPATED ERRONEOUS FPS.
23891  123436  177777                          -1              ;EXPECTED FEC.
23892  123440                          XXCDONE:
  (1)  123440  004767  001106                  JSR     PC,..RSET       ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
23893
23894
23901                                  ;*************************************************************************************
  (2)                                  ;TEST 605       STEXP TEST
  (3)                                  ;*************************************************************************************
  (2)  123444                          TS605:
23902
23903                                  ; EXP = 100 (EXCESS 200)
23904  123444  004737  123652          YYC1:   JSR     PC,@#STXSUB
23905  123450  020000  000000  000000  1$:     .WORD   20000,0,0,0                     ;AC
```

M1



```
        123456  000000
23906   123460  177700                  2$:     -100                                    ;EXP RES
23907   123462  052525                  3$:     52525                                   ;ERROR EXP.
23908   123464  040000                  4$:     40000                                   ;FPSB
23909   123466  040010                          40010                                   ;FPSA
23910   123470  040000                          40000                                   ;ERROR FPS
23911                                   ; EXP = 200 (EXCESS 200)
23912   123472  004737  123652          YYC2:   JSR     PC,@#STXSUB     ;GO EXECUTE THE INSTRUCTION.
23913   123476  040000  000000  000000  1$:     .WORD   40000,0,0,0     ;ACO OPERAND.
        123504  000000
23914   123506  000000                  2$:     0                       ;EXPECTED EXPONENT RESULT.
23915   123510  052525                  3$:     52525           ;ANTICIPATED ERRONEOUS RESULT.
23916   123512  040000                  4$:     40000                   ;FPS BEFORE EXECUTION.
23917   123514  040004                          40004                   ;FPS AFTER EXECUTION.
23918   123516  040000                          40000                   ;ANTICIPATED ERRONEOUS FPS.
23919                                   ; EXP = 201 (EXCESS 200)
23920   123520  004737  123652          YYC3:   JSR     PC,@#STXSUB     ;GO EXECUTE THE INSTRUCTION.
23921   123524  040200  000000  000000  1$:     .WORD   40200,0,0,0     ;ACO OPERAND.
        123532  000000
23922   123534  000001                  2$:     1                       ;EXPECTED EXPONENT RESULT.
23923   123536  052525                  3$:     52525           ;ANTICIPATED ERRONEOUS RESULT.
23924   123540  040000                  4$:     40000                   ;FPS BEFORE EXECUTION.
23925   123542  040000                          40000                   ;FPS AFTER EXECUTION.
23926   123544  040004                          40004                   ;ANTICIPATED ERRONEOUS FPS.
23927                                   ; EXP = 375  (EXCESS 200)
23928
23929   123546  004737  123652          YYC4:   JSR     PC,@#STXSUB     ;GO EXECUTE THE INSTRUCTION.
23930   123552  077200  000000  000000  1$:     .WORD   77200,0,0,0     ;ACO OPERAND.
        123560  000000
23931   123562  000175                  2$:     175                     ;EXPECTED EXPONENT RESULT.
23932   123564  052525                  3$:     52525           ;ANTICIPATED ERRONEOUS RESULT.
23933   123566  040000                  4$:     40000                   ;FPS BEFORE EXECUTION.
23934   123570  040000                          40000                   ;FPS AFTER EXECUTION.
23935   123572  040010                          40010                   ;ANTICIPATED ERRONEOUS FPS.
23936                                   ; EXP = 1  (EXCESS 200)
23937
23938   123574  004737  123652          YYC5:   JSR     PC,@#STXSUB     ;GO EXECUTE THE INSTRUCTION.
23939   123600  000200  000000  000000  1$:     .WORD   200,0,0,0       ;ACO OPERAND.
        123606  000000
23940   123610  177601                  2$:     -177                    ;EXPECTED EXPONENT RESULT.
23941   123612  052525                  3$:     52525           ;ANTICIPATED ERRONEOUS RESULT.
23942   123614  040000                  4$:     40000                   ;FPS BEFORE EXECUTION.
23943   123616  040010                          40010                   ;FPS AFTER EXECUTION.
23944   123620  040000                          40000                   ;ANTICIPATED ERRONEOUS FPS.
23945                                   ; EXP = 156  (EXCESS 200)
23946
23947   123622  004737  123652          YYC6:   JSR     PC,@#STXSUB     ;GO EXECUTE THE INSTRUCTION.
23948   123626  033400  000000  000000  1$:     .WORD   33400,0,0,0     ;ACO OPERAND.
        123634  000000
23949   123636  177756                  2$:     -22                     ;EXPECTED EXPONENT RESULT.
23950   123640  052525                  3$:     52525           ;ANTICIPATED ERRONEOUS RESULT.
23951   123642  047707                  4$:     47707                   ;FPS BEFORE EXECUTION.
23952   123644  047710                          47710                   ;FPS AFTER EXECUTION.
23953   123646  177777                          -1                      ;ANTICIPATED ERRONEOUS FPS.
23954   123650  000452                          BR      YYCDONE
23955
```

```
23956                                   ;THIS SUBROUTINE, STXSUB, IS USED TO SET UP THE OPERANDS, EXECUTE
23957                                   ;THE STEXP INSTRUCTION AND CHECK THE RESULTS. A CALL
23958                                   ;TO IT IS MADE THUS:
23959                                   ;
23960                                   ;               JSR     PC,@#STXSUB
23961                                   ;               ACARG:  .WORD   X,X,X,X         ;AC OPERAND
23962                                   ;               RES:    .WORD   X               ;EXPECTED RESULT
23963                                   ;               ERRES:  .WORD   X               ;ERROR RESULT
23964                                   ;               FPSB:   .WORD   X               ;FPS BEFORE EXECUTION
23965                                   ;               FPSA:   .WORD   X               ;FPS AFTER EXECUTION
23966                                   ;               ERFPS:  .WORD   X               ;ERROR FPS.
23967                                   ;               ERR1:   ERROR   X               ;DATA ERROR.
23968                                   ;                       BR      CONT
23969                                   ;               ERR2:   ERROR   X               ;FPS ERROR.
23970                                   ;               CONT:                           ;RETURN ADDRESS
23971                                   ;
23972                                   ;THE OPERANDS ARE SET UP (USING AC0 AS THE ACCUMULATOR). THEN
23973                                   ;THE STEXP INSTRUCTION IS EXECUTED.
23974                                   ;THE RESULT IS CHECKED AGAINST RES. IF THE RESULT IS CORRECT THEN THE FPS IS
23975                                   ;COMPARED WITH FPSA IF THIS TOO IS CORRECT STXSUB RETURNS CONTROL
23976                                   ;TO THE CALLING ROUTINE AT CONT. IF THE FPS IS BAD STXSUB
23977                                   ;COMPARE IT TO ERROR FPS. IF THIS MATCHES THEN STXSUB WILL RETURN
23978                                   ;TO THE ERROR CALL AT ERR2, OTHERWISE STXSUB ITSELF
23979                                   ;REPORTS THIS FAILURE AND THEN RETURNS TO CONT. IF THE RESULT OF THE
23980                                   ;STEXP IS INCORRECT, THE INCORRECT RESULT IS COMPARED WITH THE
23981                                   ;ANTICIPATED FAILING DATA PATTERN, ERRES. IF THE FAILURE IN
23982                                   ;THE RESULT WAS ANTICIPATED CORRECTLY TO BE ERRES THEN STXSUB
23983                                   ;WILL TRANSFER CONTROL TO THE ERROR CALL AT ERR1. OTHERWISE THE
23984                                   ;RESULT WAS INCORRECT BUT WAS NOT ANTICIPATED AND STXSUB WILL
23985                                   ;REPORT THE FAILURE AFTER WHICH CONTROL WILL BE PASSED TO CONT.
23986
23987  123652  012601                  STXSUB: MOV     (SP)+,R1        ;GET A POINTER TO THE ARGUMENTS.
23988  123654  010102                          MOV     R1,R2
23989  123656  012737  123456  123764          MOV     #123456,@#STXBF
23990  123664  012737  076543  123766          MOV     #76543,@#STXBF+2
23991  123672  012700  000200                  MOV     #200,R0
23992  123676  170100                          LDFPS   R0
23993  123700  010100                          MOV     R1,R0           ;SET UP THE AC0 OPERAND.
23994  123702  172410                          LDD     (R0),AC0
23995  123704  016100  000016                  MOV     16(R1),R0       ;SET THE FPS.
23996  123710  170100                          LDFPS   R0
23997  123712  012700  123764                  MOV     #STXBF,R0
23998  123716  175010                  1$:     STEXP   AC0,(R0)        ;TEST INSTRUCTION.
23999  123720  170204                          STFPS   R4              ;GET FPS.
24000  123722  026137  000010  123764          CMP     10(R1),@#STXBF  ;WAS RESULT CORRECT?
24001  123730  001401                          BEQ     5$
  (2)  123732  104000                          EMT                     ;
24002  123734  020461  000016                  CMP     R4,16(R1)       ;SEE IF THE FPS IS CORRECT.
24003  123740  001401                          BEQ     10$
  (2)  123742  104000                          EMT                     ;
24004                                   ;SEE IF MORE THAN ONE WORD WAS WRITTEN IN THE OUTPUT BUFFER.
24005  123744  022737  076543  123766  10$:    CMP     #76543,@#STXBF+2
24006  123752  001401                          BEQ     4$
  (2)  123754  104000                          EMT                     ;
24007  123756  000161  000022          4$:     JMP     22(R1)
24008
```

```
24009  123762  177777                          -1
24010  123764  177777  177777  177777  STXBF:  .WORD   -1,-1,-1,-1,-1
       123772  177777  177777
24011
24012  123776                          YYCDONE:
  (1)  123776  004767  000550                  JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
24013
24024                                  ;*****************************************************************************
  (2)                                  ;TEST 606       STST TEST
  (3)                                  ;*****************************************************************************
  (2)  124002                          TS606:
24025
24026  124002  012700  040000                  MOV     #40000,R0       ;SET FPS. FID=1.
24027  124006  170100                          LDFPS   R0
24028
24029  124010  170003                  ZZC2:   .WORD   170003                  ;ILLEGAL FPP
24030                                                                    ;OP CODE
24031  124012  012700  124072                  MOV     #ZZCBF,R0       ;SET UP THE OUTPUT BUFFER.
24032  124016  012710  177777                  MOV     #-1,(R0)
24033  124022  012760  177777  000002          MOV     #-1,2(R0)
24034  124030  170310                  ZZC3:   STST    (R0)                    ;GET FEC AND
24035                                                                    ;FEA
24036  124032  170204                          STFPS   R4              ;GET FPS.
24037  124034  012700  124072                  MOV     #ZZCBF,R0
24038  124040  022710  000002                  CMP     #2,(R0) ;SEE IF FEC IS CORRECT.
24039  124044  001010                          BNE     ZZC10           ;BRANCH IF INCORRECT.
24040  124046  022760  124010  000002          CMP     #ZZC2,2(R0)     ;SEE IF FEA, ADDRESS, IS CORRECT.
24041  124054  001004                          BNE     ZZC10           ;BRANCH IF INCORRECT.
24042  124056  022704  140000                  CMP     #140000,R4      ;SEE IF FPS IS CORRECT.
24043  124062  001001                          BNE     ZZC10           ;BRANCH IF INCORRECT.
24044  124064  000407                          BR      ZZCDONE
24045  124066                          ZZC10:
  (2)  124066  104000                          EMT                     ;
24046
24047                                  ;DATA BUFFER:
24048  124070  177777                          -1
24049  124072  177777  177777  177777  ZZCBF:  .WORD   -1,-1,-1,-1
       124100  177777
24050  124102  177777                          -1
24051
24052  124104                          ZZCDONE:
  (1)  124104  004767  000442                  JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                  ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                  ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                  ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                  ;THE USER TYPED CONTROL G?).
24053
24054
24055                                  ;*****************************************************************************
  (2)                                  ;TEST 607       SPECIAL CASE TEST
  (3)                                  ;*****************************************************************************
  (2)  124110                          TS607:
```

```
24056  124110  012746  144724          AAD1:   MOV     #144724, -(SP)  ;PUT FRACTION ON STACK
24057  124114  012746  040600                  MOV     #40600, -(SP)   ;PUT EXPONENT ON STACK
24058  124120  005046                          CLR     -(SP)           ;PUT SUBTRAHEND FRACTION ON STACK
24059  124122  012746  040600                  MOV     #40600, -(SP)   ;PUT SUBTRAHEND EXPONENT ON STACK
24060  124126  172466  000004                  LDF     4(SP), AC0      ;LOAD FP ACCUMULATORS
24061  124132  173026                          SUBF    (SP)+, AC0      ;DO SUBTRACTION
24062  124134  174037  124164                  STF     AC0, @#AADBF    ;GET AND STORE ANSWER
24063  124140  022737  036711  124164          CMP     #36711, @#AADBF ;IS EXPONENT CORRECT
24064  124146  001401                          BEQ     1$
  (2)  124150  104000                          EMT                     ;BAD EXPONENT FROM SUBTRACTION
24065  124152  022737  152000  124166  1$:     CMP     #152000, @#AADBF+2 ;IS FRACTION CORRECT
24066  124160  001403                          BEQ     AADDONE
  (2)  124162  104000                          EMT                     ;FRACTION INCORRECT
24067
24068  124164  000000          AADBF:  .WORD   0
24069  124166  000000                  .WORD   0
24070
24071  124170  012706  001000          AADDONE: MOV    #STBOT, SP      ;RESTORE STACK POINTER
24072  124174  004767  000352                  JSR     PC,.RSET        ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                   ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                   ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                   ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                   ;THE USER TYPED CONTROL G?).
24073
24074
24075
24076                                   ;*****************************************************************************
  (2)                                   ;TEST 610       INTERRUPTABILITY TEST
  (3)                                   ;*****************************************************************************
  (2)  124200                           TS610:
24077  124200  004567  124416          BBD1:   JSR     R5,CHKAPT
24078  124204  000520                          BR      FPEXIT          ;SKIP TEST IF ON APT AND NOT FIRST PASS
24079
24080  124206  005001                          CLR     R1              ;INITIALIZE A COUPLE OF COUNTERS
24081  124210  005000                          CLR     R0
24082  124212  013767  000064  113352          MOV     @#64,   $TMP0   ;SAVE INTERRUPT VECTOR
24083  124220  013767  000066  113346          MOV     @#66,   $TMP1   ;SAVE INTERRUPT PRIORITY
24084  124226  012737  124300  000064          MOV     #3$,    @#64    ;SET UP INTERRUPT PRIORITY FOR THIS TEST
24085  124234  005037  000066                  CLR     @#66            ;AND PRIORITY
24086  124240  005067  053532                  CLR     PS              ;PUT PROCESSOR PRIORITY AT 0
24087  124244  005067  053316                  CLR     TPB             ;SEND A NULL CHARACTER
24088  124250  105767  053310          1$:     TSTB    TTCSR           ;WAIT FOR DONE TO SET
24089  124254  100375                          BPL     1$
24090  124256  005067  053304                  CLR     TPB             ;SEND A SECOND CHARACTER
24091  124262  052767  000100  053274          BIS     #BIT6,  TTCSR   ;SET INTERRUPT ENABLE
24092  124270  005200                  2$:     INC     R0              ;INCREMENT COUNTER TO GET BASE TIME
24093  124272  001376                          BNE     2$              ;CONTINUE LOOPING UNLESS COUNTER GOES TO 0
24094  124274  000005                          RESET                   ;IF NO INTERRUPT YET KILL IT
24095  124276  104000                          EMT                     ;NO INTERRUPT OCCURRED IN ALLOTTED TIME
24096  124300  166700  000110          3$:     SUB     Y,      R0      ;SUBTRACT TIME FOR FP INSTRUCTION
24097  124304  010067  000106                  MOV     R0,     Z       ;SAVE FIRST TIME
24098  124310  012737  124366  000064          MOV     #7$,    @#64    ;SET UP FOR NEXT INTERRUPT
24099  124316  005100                  4$:     COM     R0              ;MAKE PRE LOOP COUNTER NEGATIVE
24100  124320  005067  053240                  CLR     TTCSR           ;MAKE SURE NO INTERRUPT YET
24101  124324  005067  053236                  CLR     TPB             ;SEND A CHARACTER
24102  124330  105767  053230          5$:     TSTB    TTCSR           ;WAIT FOR READY BIT TO SET
```

```
24103  124334  100375                        BPL     5$
24104  124336  005067  053224                CLR     TPB                     ;SEND SECOND CHARACTER
24105  124342  052767  000100  053214        BIS     #BIT6,  TTCSR           ;SET INTERRUPT ENABLE
24106  124350  005200                6$:     INC     R0                      ;DO PRE LOOP
24107  124352  001376                        BNE     6$
24108  124354  171227  040400                MULF    #2.,    AC2             ;DO FLOATING POINT INSTRUCTION
24109  124360  000240                        NOP                             ;JUST IN CASE INTERRUPT TAKES TOO LONG
24110  124362  000005                        RESET                           ;IF NO INTERRUPT CLEAR THE WORLD
24111  124364  104000                        EMT                             ;INTERRUPT NOT BACK IN ALLOTTED TIME
24112  124366  005201                7$:     INC     R1                      ;INCREMENT TIMES THROUGH COUNTER
24113  124370  020127  000015                CMP     R1,     #15             ;HAVE WE PASSED HERE 15 TIMES BEFORE
24114  124374  001411                        BEQ     BBDDONE                 ;IF YES I MAY NEVER PASS HERE AGAIN
24115  124376  062767  000002  000012        ADD     #2.,    Z               ;IF NO ADD A LITTLE TIME TO PRELOOP
24116  124404  016700  000006                MOV     Z,      R0              ;PUT NEW COUNT IN COUNTER
24117  124410  000742                        BR      4$                      ;DO IT ALL AGAIN
24118
24119  124412  000000                X:      .WORD   0
24120  124414  000026                Y:      .WORD   26
24121  124416  000000                Z:      .WORD   0
24122
24123  124420  042767  000100  053136  BBDDONE:BIC     #100,   TTCSR           ;CLEAR INTERRUPT ENABLE BEFORE EXITING TEST
24124  124426  016737  113140  000064        MOV     $TMP0,  @#64            ;RESTORE PRINTER VECTOR
24125  124434  016737  113134  000066        MOV     $TMP1,  @#66            ;RESTORE PRINTER PRIORITY
24126  124442  004767  000104                JSR     PC,..RSET               ;GO INITIALIZE THE FPS AND STACK; AND
  (1)                                                                       ;SEE IF THE USER HAS EXPRESSED
  (1)                                                                       ;THE DESIRE TO CHANGE THE SOFTWARE
  (1)                                                                       ;VIRTUAL CONSOLE SWITCH REGISTER (HAS
  (1)                                                                       ;THE USER TYPED CONTROL G?).
24127  124446  000167  000164        FPEXIT: JMP     SLU1ST                  ;GET OVER SUBROUTINES TO NEXT TEST
24128
24129
24130  124452  004767  006752        ERROR4: JSR     PC,ABORT                ;ARE WE UNDER UFD ?
24131  124456  012737  000003  001002        MOV     #3,@#$FATAL             ;SET UP FATAL ERROR NUMBER
24132  124464  012767  000001  054306        MOV     #1,$MSGTY               ;SET FATAL ERROR FLAG
24133  124472  032737  000001  001020        BIT     #1,@#ENV                ;UNDER APT
24134  124500  001004                        BNE     FPHLT                   ;YES
24135  124502  012700  124514                MOV     #FPMSG,R0
24136  124506  004767  006652                JSR     PC,TYPE
24137  124512  000777                FPHLT:  BR      .                       ;STAY HERE FOREVER
24138
24139  124514  040506  046111  042105  FPMSG:  .ASCIZ  /FAILED DURING THE FPP TESTS/<12><15>
       124522  042040  051125  047111
       124530  020107  044124  020105
       124536  050106  020120  042524
       124544  052123  005123  000015
24140                                   .EVEN
24141
24142                                           .SBTTL  FLAG RESET ROUTINE
24143                                   ;*******************************************************************************
24144                                   ;*THIS ROUTINE WILL BE CALLED AT THE END OF EACH FLOATING POINT TEST
24145                                   ;*TO RESET THE STACK, CLEAR THE FPS AND REINITIALIZE TRAP VECTORS
24146
24147  124552  012737  124452  000244  .RSET:  MOV     #ERROR4,@#FPVECT
24148  124560  012737  021526  000004        MOV     #TO4,@#ERRVECT
24149  124566  012737  021530  000010        MOV     #TO10,@#10
24150  124574  011600                        MOV     (SP),R0
```

```
24151  124576  012706  001000                  MOV     #STBOT,SP
24152  124602  005004                          CLR     R4
24153  124604  170104                          LDFPS   R4
24154  124606  000110                          JMP     (R0)
24155
24156                                  ;THESE ARE SOME EQUATES USED IN THE  PROGRAM
24157          000001                          BIT0=000001
24158          000002                          BIT1=000002
24159          000004                          BIT2=000004
24160          000010                          BIT3=000010
24161          000020                          BIT4=000020
24162          000040                          BIT5=000040
24163          000100                          BIT6=000100
24164          000200                          BIT7=000200
24165          000400                          BIT8=000400
24166          001000                          BIT9=001000
24167          002000                          BIT10=002000
24168          004000                          BIT11=004000
24169          010000                          BIT12=010000
24170          020000                          BIT13=020000
24171          040000                          BIT14=040000
24172          100000                          BIT15=100000
24173
24174  124610  177560                  RCSR:   177560                  ;ADDRESS OF RECEIVER COMMAND/STATUS REGISTER
24175  124612  177562                  RBUF:   177562                  ;ADDRESS OF RECEIVER BUFFER
24176  124614  177564                  TCSR:   177564                  ;ADDRESS OF TRANSMITTER COMMAND/STATUS REGISTER
24177  124616  177566                  TBUF:   177566                  ;ADDRESS OF TRANSMITTER BUFFER
24178  124620  000060                  RVECT:  60                      ;RECEIVER INTERRUPT VECTOR
24179  124622  000062                  RPSW:   62
24180  124624  000064                  TVECT:  64                      ;TRANSMITTER INTERRUPT VECTOR
24181  124626  000066                  TPSW:   66
24182
24183                                  ;REAL TIME CLOCK REGISTER AND VECTOR ADDRESSES
24184  124630  177546                  LKS:    .WORD   177546
24185  124632  000100                  RTCVT:  .WORD   100
24186  124634  000102                  RTCPSW: .WORD   102
24187
24188  124636  000244                  SLU1ST: CLZ
24189  124640  032777  000004  074730          BIT     #4,@SWR
24190  124646  001402                          BEQ     1$
24191  124650  000167  001112                  JMP     KWSTRT
24192  124654  012737  000004  001004  1$:     MOV     #4,@#TESTN      ;PUT TEST NUMBER IN MAILBOX
24193  124662  012737  125700  000030          MOV     #ERROR5,@#30    ;SET UP FOR CORRECT ERROR CALL
24194
24195
24196                                  ;******************************************************************************
  (2)                                 ;TEST 611       TEST ABILITY TO REFERENCE TCSR
  (3)                                 ;******************************************************************************
  (2)   124670                          TS611:
24197  124670  013703  000004                  MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
24198  124674  012737  124710  000004          MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
24199  124702  005777  177706                  TST     @TCSR           ;REFERENCE THE XMIT COMMAND/STATUS REG.
24200  124706  000401                          BR      4$
  (1)   124710                          1$:
  (2)   124710  104000                          EMT                     ;
24201  124712  010337  000004          4$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
```

```
                                        ;*******************************************************************************
                                        ;TEST 612       TEST ABILITY TO REFERENCE TBUF
                                        ;*******************************************************************************
124716                                  TS612:
124716  013703  000004                          MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
124722  012737  124736  000004                  MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
124730  005777  177662                          TST     @TBUF           ;REFERENCE THE XMIT BUFFER
124734  000401                                  BR      4$
124736                                  1$:
124736  104000                                  EMT                     ;
124740  010337  000004                  4$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR


                                        ;*******************************************************************************
                                        ;TEST 613       TEST THAT TCSR BIT7(DONE) CLEARS WHEN XBUF IS LOADED
                                        ;*******************************************************************************
124744                                  TS613:
124744  032737  000001  001020                  BIT     #1,@#$ENV       ;ARE WE RUNNING UNDER APT
124752  001405                                  BEQ     70$             ;IF NO THEN SERIES OF TESTS
124754  005737  001006                          TST     @#$PASS         ;IS THIS FIRST PASS
124760  001402                                  BEQ     70$             ;IF YES THEN DO SERIES OF TESTS
124762  000167  001000                          JMP     KWSTRT          ;IF NO THEN BYPASS SERIES OF TESTS
124766  005077  177624                  70$:    CLR     @TBUF           ;LOAD XBUF
124772  105777  177616                          TSTB    @TCSR           ;CHECK DONE
124776  100006                                  BPL     3$              ;BR IF CLEAR
                                                                        ;FILL SECOND BUFFER BECUASE REFRESH COULD CAUSE
                                                                        ;FIRST TEST TO FAIL
125000  005077  177612                          CLR     @TBUF           ;FILL DOUBLE BUFFER
125004  105777  177604                          TSTB    @TCSR           ;CHECK DONE
125010  100001                                  BPL     3$
125012  104000                                  EMT                     ;
125014  005000                          3$:     CLR     R0              ;CLEAR TIMER
125016  105777  177572                  4$:     TSTB    @TCSR           ;CHECK FOR XMIT DONE
125022  100403                                  BMI     5$              ;IF DONE SETS, BR TO END OF TEST
125024  005200                                  INC     R0              ;INCREMENT TIMER
125026  001373                                  BNE     4$
125030  104000                                  EMT                     ;
125032                                  5$:


                                        ;*******************************************************************************
                                        ;TEST 614       TEST THAT TCSR "DONE" SETS WITH RESET
                                        ;*******************************************************************************
125032                                  TS614:
125032  005077  177560                          CLR     @TBUF           ;LOAD TRANSMIT BUFFER
125036  105777  177552                  1$:     TSTB    @TCSR           ;WAIT FOR DONE
125042  100375                                  BPL     1$
125044  005077  177546                          CLR     @TBUF           ;LOAD SECOND BUFFER
125050  000240                                  NOP
125052  000005                                  RESET                   ;SET DONE WITH RESET
125054  105777  177534                          TSTB    @TCSR           ;CHECK FOR DONE SET
125060  100401                                  BMI     TS615
125062  104000                                  EMT                     ;
```

```
                                        ;*******************************************************************************
                                        ;TEST 615        TEST ABILITY TO ACCESS RCSR
                                        ;*******************************************************************************
125064                                  TS615:
125064  013703  000004                          MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
125070  012737  125104  000004                  MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
125076  005777  177506                          TST     @RCSR           ;ACCESS RCSR
125102  000401                                  BR      2$
125104                                  1$:
125104  104000                                  EMT                     ;
125106  010337  000004                  2$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR


                                        ;*******************************************************************************
                                        ;TEST 616        TEST ABILITY TO ACCESS RBUF
                                        ;*******************************************************************************
125112                                  TS616:
125112  013703  000004                          MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
125116  012737  125132  000004                  MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
125124  005777  177462                          TST     @RBUF           ;ACCESS RBUF
125130  000401                                  BR      2$
125132                                  1$:
125132  104000                                  EMT                     ;
125134  010337  000004                  2$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR



                                        ;*******************************************************************************
                                        ;TEST 617        TEST THAT BIT6 OF RCSR CAN BE SET & RESET
                                        ;*******************************************************************************
125140                                  TS617:
125140  017703  177454                          MOV     @RVECT,R3       ;SAVE RECEIVE VECTOR
125144  012777  125166  177446                  MOV     #1$,@RVECT      ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
125152  106427  000340                          MTPS    #340            ;SET PSW TO PRIORITY 7
125156  032777  000100  177424                  BIT     #BIT6,@RCSR     ;TEST BIT6 OF RCSR
125164  001401                                  BEQ     2$
125166                                  1$:
125166  104000                                  EMT                     ;
125170  052777  000100  177412          2$:     BIS     #BIT6,@RCSR     ;SET BIT6 OF RCSR
125176  032777  000100  177404                  BIT     #BIT6,@RCSR     ;TEST BIT6 OF RCSR
125204  001001                                  BNE     3$
125206  104000                                  EMT                     ;
125210  042777  000100  177372          3$:     BIC     #BIT6,@RCSR     ;CLEAR BIT6 OF RCSR
125216  032777  000100  177364                  BIT     #BIT6,@RCSR     ;TEST BIT6 OF RCSR
125224  001401                                  BEQ     4$
125226  104000                                  EMT                     ;
125230                                  4$:
125230  052777  000100  177352                  BIS     #BIT6,@RCSR     ;SET BIT6 OF RCSR
125236  000005                                  RESET                   ;CLEAR BIT6OF RCSR WITH RESET
125240  032777  000100  177342                  BIT     #BIT6,@RCSR     ;TEST BIT6 OF RCSR
125246  001401                                  BEQ     5$
125250  104000                                  EMT                     ;
```

```
24282  125252  010377  177342          5$:     MOV     R3,@RVECT       ;RESTORE RECEIVE VECTOR
24283
24284
24285
24286                                  ;*************************************************************************************
  (2)                                  ;TEST 620       TEST THAT XMIT INTERRUPTS ONLY WHEN ENABLED
  (3)                                  ;*************************************************************************************
  (2)   125256                         TS620:
24287   125256  042777  000100  177330          BIC     #BIT6,@TCSR     ;CLEAR TRANSMIT INTERRUPT ENABLE
24288   125264  017703  177334                  MOV     @TVECT,R3       ;SAVE XMIT VECTOR
24289   125270  012777  125312  177326          MOV     #2$,@TVECT      ;POINT XMIT VECTOR TO ERROR REPORT
24290   125276  105777  177312          1$:     TSTB    @TCSR           ;WAIT FOR DONE
24291   125302  100375                  BPL     1$
24292   125304  106427  000140          MTPS    #140            ;SET PSW TO PRIORITY 3
24293   125310  000401                  BR      3$
  (1)   125312                         2$:
  (2)   125312  104000                  EMT                     ;
24294   125314  012777  125334  177302  3$:     MOV     #4$,@TVECT      ;SET XMIT VECTOR TO END OF TEST
24295   125322  052777  000100  177264          BIS     #BIT6,@TCSR     ;ENABLE INTERRUPTS
24296   125330  000240                  NOP
24297
24298   125332  104000                  EMT                     ;XMIT DID NOT INTERRUPT
24299
24300   125334  042777  000100  177252  4$:     BIC     #BIT6,@TCSR     ;DISABLE INTERRUPTS
24301   125342  022626                  CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
24302   125344  010377  177254                  MOV     R3,@TVECT       ;RESTORE XMIT VECTOR
24303
24304
24305                                  ;*************************************************************************************
  (2)                                  ;TEST 621       TEST THAT XMIT INTERRUPTS DO NOT OCCUR WHEN DISABLED
  (3)                                  ;*************************************************************************************
  (2)   125350                         TS621:
24306   125350  042777  000100  177236          BIC     #BIT6,@TCSR     ;DISABLE INTERRUPTS
24307   125356  106427  000340                  MTPS    #340            ;SET PSW TO PRIORITY 7
24308   125362  017703  177236                  MOV     @TVECT,R3       ;SAVE XMIT VECTOR
24309   125366  012777  125414  177230          MOV     #2$,@TVECT      ;POINT XMIT VECTOR TO ERROR REPORT
24310   125374  105777  177214          1$:     TSTB    @TCSR           ;WAIT FOR DONE
24311   125400  100375                  BPL     1$
24312   125402  052777  000100  177204          BIS     #BIT6,@TCSR     ;ENABLE INTERRUPT
24313   125410  000240                  NOP
24314   125412  000401                  BR      3$
  (1)   125414                         2$:
  (2)   125414  104000                  EMT                     ;
24315   125416  042777  000100  177170  3$:     BIC     #BIT6,@TCSR     ;CLEAR INTERRUPT ENABLE
24316   125424  012777  125442  177172          MOV     #4$,@TVECT      ;POINT XMIT VECTOR TO ERROR REPORT
24317   125432  106427  000140                  MTPS    #140            ;SET PSW TO PRIORITY 3
24318   125436  000240                  NOP
24319   125440  000401                  BR      5$
  (1)   125442                         4$:
  (2)   125442  104000                  EMT                     ;
24320   125444  010377  177154          5$:     MOV     R3,@TVECT       ;RESTORE XMIT VECTOR
24321
24322
24323                                  ;*************************************************************************************
  (2)                                  ;TEST 622       TEST TRANSMITTER FOR DOUBLE INTERRUPTS
  (3)                                  ;*************************************************************************************
```

```
  (2)  125450                                TS622:
24324  125450  042777  000100  177136                BIC     #BIT6,@TCSR      ;CLEAR INTERRUPT ENABLE
24325  125456  017703  177142                        MOV     @TVECT,R3        ;SAVE XMIT VECTOR
24326  125462  017704  177140                        MOV     @TPSW,R4         ;SAVE XMIT PSW VECTOR
24327  125466  012777  125526  177130                MOV     #2$,@TVECT       ;SET UP XMIT VECTOR
24328  125474  012777  000340  177124                MOV     #340,@TPSW       ;SET PIO 7 AFTER INTERRUPT
24329  125502  106427  000140                        MTPS    #140             ;SET PSW TO PRIORITY 3
24330  125506  105777  177102                1$:     TSTB    @TCSR            ;WAIT FOR DONE
24331  125512  100375                                BPL     1$
24332  125514  052777  000100  177072                BIS     #BIT6,@TCSR      ;ENABLE INTERRUPTS
24333  125522  000240                                NOP
24334
24335  125524  104000                                EMT                      ;
24336                                                                          ;XMIT INTERRUPT DID NOT OCCUR
24337  125526  022626                        2$:     CMP     (SP)+,(SP)+      ;RESTORE SP AFTER INTERRUPT
24338  125530  012777  125554  177066                MOV     #4$,@TVECT       ;POINT XMIT VECTOR TO ERROR
24339  125536  106427  000140                        MTPS    #140             ;SET PSW TO PRIORITY 3
24340  125542  000240                                NOP                      ;GIVE TIME FOR ANY INTERRUPTS
24341  125544  042777  000100  177042                BIC     #BIT6,@TCSR      ;DISABLE INTERRUPTS
24342  125552  000401                                BR      5$
  (1)  125554                                4$:
  (2)  125554  104000                                EMT                      ;
24343  125556  010377  177042                5$:     MOV     R3,@TVECT        ;RESTORE XMIT VECTOR
24344  125562  010477  177040                        MOV     R4,@TPSW         ;RESTORE XMIT PSW VECTOR
24345
24346                                        ;*******************************************************************************
  (2)                                        ;TEST 623       TEST THAT XMIT INTERRUPT CLEARS WITH LOADING TBUF
  (3)                                        ;*******************************************************************************
  (2)  125566                                TS623:
24347  125566  042777  000100  177020                BIC     #BIT6,@TCSR      ;DISABLE INTERRUPTS
24348  125574  106427  000340                        MTPS    #340             ;SET PSW TO PRIORITY 7
24349  125600  017703  177020                        MOV     @TVECT,R3        ;SAVE XMIT VECTOR
24350  125604  012777  125654  177012                MOV     #2$,@TVECT       ;POINT XMIT VECTOR TO ERROR
24351  125612  052777  000100  176774                BIS     #BIT6,@TCSR      ;ENABLE INTERRUPTS
24352  125620  005077  176772                        CLR     @TBUF            ;LOAD TBUF
24353  125624  105777  176764                1$:     TSTB    @TCSR            ;WAIT FOR DONE (INTERRUPT)
24354  125630  100375                                BPL     1$
24355  125632  005077  176760                        CLR     @TBUF            ;FILL SECOND BUFFER TO RESET INT.
24356  125636  106427  000140                        MTPS    #140             ;SET PSW TO PRIORITY 3
24357  125642  000240                                NOP                      ;GIVE TIME FOR ANY INTERRUPTS
24358  125644  042777  000100  176742                BIC     #BIT6,@TCSR      ;DISABLE INTERRUPTS
24359  125652  000401                                BR      3$
  (1)  125654                                2$:
  (2)  125654  104000                                EMT                      ;
24360  125656  010377  176742                3$:     MOV     R3,@TVECT        ;RESTORE XMIT VECTOR
24361  125662  005000                                CLR     R0               ;INITIALIZE LOOP COUNTER
24362  125664  005200                        4$:     INC     R0               ;INCREMENT LOOP COUNTER
24363  125666  001376                                BNE     4$               ; UNTIL COUNTER = 0
24364  125670  005777  176716                        TST     @RBUF            ;CLEAR RECEIVER BUFFER
24365  125674  000167  000066                        JMP     KWSTRT           ;GET TO NEXT TEST
24366
24367  125700  004767  005524                ERROR5: JSR     PC,ABORT         ;ARE WE UNDER UFD ?
24368  125704  012737  000004  001002                MOV     #4,@#$FATAL      ;SET UP FATAL ERROR NUMBER
24369  125712  012767  000001  053060                MOV     #1,$MSGTY        ;SET FATAL ERROR FLAG
24370  125720  032737  000001  001020                BIT     #1,@#$ENV                        ;UNDER APR ?
24371  125726  001004                                BNE     SL1HLT
```

```
 24372  125730  012700  125742                  MOV     #SL1MSG,R0
 24373  125734  004767  005424                  JSR     PC,TYPE
 24374  125740  000777                  SL1HLT: BR      .
 24375
 24376  125742  040506  046111  042105  SL1MSG: .ASCIZ  /FAILED SLU1 TEST/<12><15>
        125750  051440  052514  020061
        125756  042524  052123  006412
        125764     000
 24377          125766                  .EVEN
 24378
 24379
 24380  125766  000244                  KWSTRT: CLZ
 24381  125770  032777  000010  073600          BIT     #10,@SWR
 24382  125776  001402                          BEQ     1$
 24383  126000  000167  000734                  JMP     SLU2ST
 24384  126004  012737  000005  001004  1$:     MOV     #5,@#$TESTN     ;PUT TEST NUMBER IN MAILBOX
 24385  126012  012737  126624  000030          MOV     #ERROR6,@#30    ;SET UP ERROR CALL
 24386
 24387  126020                          LKSTST:
 24388                                  ;*****************************************************************************
   (2)                                  ;TEST 624       TEST ABILITY TO ACCESS LKS
   (3)                                  ;*****************************************************************************
   (2)  126020                          TS624:
 24389  126020  013703  000004                  MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
 24390  126024  012737  126040  000004          MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
 24391  126032  005777  176572                  TST     @LKS            ;ACCESS LKS
 24392  126036  000401                          BR      2$
   (1)  126040                          1$:
   (2)  126040  104000                          EMT                     ;
 24393  126042  010337  000004          2$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
 24394
 24395                                  ;*****************************************************************************
   (2)                                  ;TEST 625       TEST THAT BIT6 OF LKS CAN BE SET & RESET
   (3)                                  ;*****************************************************************************
   (2)  126046                          TS625:
 24396  126046  017703  176560                  MOV     @RTCVT,R3       ;SAVE LINE CLOCK VECTOR
 24397  126052  012777  126074  176552          MOV     #1$,@RTCVT      ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
 24398  126060  106427  000340                  MTPS    #340            ;SET PSW TO PRIORITY 7
 24399  126064  032777  000100  176536          BIT     #BIT6,@LKS      ;TEST BIT6 OF LKS
 24400  126072  001401                          BEQ     2$
   (1)  126074                          1$:
   (2)  126074  104000                          EMT                     ;
 24401  126076  052777  000100  176524  2$:     BIS     #BIT6,@LKS      ;SET BIT6 OF LKS
 24402  126104  032777  000100  176516          BIT     #BIT6,@LKS      ;TEST BIT6 OF LKS
 24403  126112  001001                          BNE     3$
   (2)  126114  104000                          EMT                     ;
 24404  126116  042777  000100  176504  3$:     BIC     #BIT6,@LKS      ;CLEAR BIT6 OF LKS
 24405  126124  032777  000100  176476          BIT     #BIT6,@LKS      ;TEST BIT6 OF LK
 24406  126132  001401                          BEQ     4$
   (2)  126134  104000                          EMT                     ;
 24407  126136  032737  000001  001020  4$:     BIT     #1,@#$ENV       ;ARE WE RUNNING UNDER APT
 24408  126144  001403                          BEQ     70$             ;IF NO THEN DO TEST
 24409  126146  005737  001006                  TST     @#$PASS         ;IS THIS FIRST PASS
 24410  126152  001011                          BNE     5$              ;IF NO SKIP TO TEST END
 24411  126154                          70$:
 24412  126154  052777  000100  176446          BIS     #BIT6,@LKS      ;SET BIT6 OF LKS
```

```
24413  126162  000005                          RESET                   ;CLEAR BIT6 OF LKS WITH RESET
24414  126164  032777  000100  176436          BIT     #BIT6,@LKS      ;TEST BIT6 OF LKS
24415  126172  001401                          BEQ     5$
  (2)  126174  104000                          EMT                     ;
24416  126176  010377  176430          5$:     MOV     R3,@RTCVT       ;RESTORE LINE CLOCK VECTOR
24417
24418
24419                                  ;*************************************************************************
  (2)                                  ;TEST 626       TEST THAT THE REAL TIME CLOCK INTERRUPTS PROPERLY
  (3)                                  ;*************************************************************************
  (2)  126202                          TS626:
24420
24421  126202  106427  000340                  MTPS    #340            ;SET PSW TO PRIORITY 7
24422  126206  017703  176420                  MOV     @RTCVT,R3       ;SAVE LINE CLOCK VECTOR
24423  126212  017704  176416                  MOV     @RTCPSW,R4      ;SAVE LINE CLOCK PSW VECTOR
24424  126216  012777  126260  176406          MOV     #ESR51,@RTCVT   ;SET RTC INTERRUPT VECTOR TO ERROR REPORT
24425  126224  012777  000340  176402          MOV     #340,@RTCPSW    ;KEEP PRIORITY AT 7
24426  126232  052777  000100  176370          BIS     #BIT6,@LKS      ;SET INTERRUPT ENABLE
24427
24428  126240  012701  024000                  MOV     #24000,R1               ;SET UP A WAIT LOOP
24429  126244  077101                  ESR36:  SOB     R1,ESR36                ;WAIT 30 MILLISEC
24430  126246  012777  126262  176356          MOV     #ESR37,@RTCVT           ;ALTER VECTOR
24431  126254  106427  000240                  MTPS    #240            ;PRIORITY NOW TO FIVE
24432  126260  104000                  ESR51:  EMT
24433  126262  012777  126300  176342  ESR37:  MOV     #ESR38,@RTCVT
24434  126270  012701  024000                  MOV     #24000,R1
24435  126274  077101                  ESR39:  SOB     R1,ESR39        ;WAIT 30 MORE MILLISEC
24436  126276  000401                          BR      ESR38+2
24437  126300  104000                  ESR38:  EMT                     ;AN ERROR IF WE ARE HERE
24438  126302  005077  176322                  CLR     @LKS            ;CLR INTERRUPT ENABLE
24439  126306  012701  024000                  MOV     #24000,R1
24440  126312  077101                  ESR52:  SOB     R1,ESR52        ;WAIT LOOP
24441  126314  106427  000240                  MTPS    #240                    ;ALTER PRIORITY TO FIVE
24442  126320  012701  024000                  MOV     #24000,R1
24443  126324  077101                  ESR53:  SOB     R1,ESR53        ;WAIT AGAIN
24444  126326  012777  126352  176276          MOV     #ESR55,@RTCVT
24445  126334  052777  000100  176266          BIS     #BIT6,@LKS              ;ENABLE LTC INTERRUPTS
24446  126342  012701  024000                  MOV     #24000,R1
24447  126346  077101                  ESR54:  SOB     R1,ESR54
24448  126350  104000                          EMT                             ;SHOULD HAVE INTERRUPTED
24449
24450  126352  022626                  ESR55:  CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
24451  126354  042777  000100  176246          BIC     #BIT6,@LKS      ;DISABLE INTERRUPTS
24452  126362  010377  176244                  MOV     R3,@RTCVT       ;RESTORE LINE CLOCK VECTOR
24453  126366  010477  176242                  MOV     R4,@RTCPSW      ;RESTORE LINE CLOCK PSW VECTOR
24454
24455
24456
24457
24458
24459                                  ;*************************************************************************
  (2)                                  ;TEST 627       TEST RTC FOR DOUBLE INTERRUPTS
  (3)                                  ;*************************************************************************
  (2)  126372                          TS627:
24460  126372  032737  000001  001020          BIT     #1, @#$ENV      ;ARE WE RUNNING UNDER APT
  (1)  126400  001403                          BEQ     70$             ;IF NO THEN DO TEST
```

```
  (1)  126402  005737  001006            TST     @#$PASS          ;IS THIS FIRST PASS
  (2)  126406  001045                     BNE     TS630            ;IF NO THEN SHIP TO NEXT TEST
  (1)  126410                        70$:
24461  126410  017703  176216            MOV     @RTCVT,R3        ;SAVE LINE CLOCK VECTOR
24462  126414  017704  176214            MOV     @RTCPSW,R4       ;SAVE LINE CLOCK PSW VECTOR
24463  126420  012777  126462  176204    MOV     #2$,@RTCVT       ;SET UP RTC INTERRUPT VECTOR
24464  126426  012777  000340  176200    MOV     #340,@RTCPSW     ;DISALLOW INTERRUPTS AFTER THE INTERRUPT
24465  126434  106427  000240            MTPS    #240             ;SET PSW TO PRIORITY 5
24466
24467  126440  005000                     CLR     R0
24468  126442  052777  000100  176160    BIS     #BIT6,@LKS       ;ENABLE CLOCK INTERRUPTS
24469  126450  005200                1$:  INC     R0
24470  126452  005700                     TST     R0
24471  126454  100375                     BPL     1$
24472  126456  000240                     NOP                      ;GIVE TIME FOR ANY INTERRUPT
24473
24474  126460  104000                     EMT                      ;RTC INTERRUPT DID NOT OCCUR
24475
24476  126462  022626                2$:  CMP     (SP)+,(SP)+      ;RESTORE SP AFTER INTERRUPT
24477  126464  012777  126502  176140    MOV     #3$,@RTCVT       ;POINT RTC VECTOR TO ERROR REPORT
24478  126472  106427  000240            MTPS    #240             ;SET PSW TO PRIORITY 5
24479  126476  000240                     NOP                      ;GIVE SOME TIME FOR AN INTERRUPT
24480  126500  000401                     BR      4$
  (1)  126502                        3$:
  (2)  126502  104000                     EMT                      ;
24481  126504  042777  000100  176116  4$:  BIC     #BIT6,@LKS       ;DISABLE CLOCK INTERRUPTS
24482  126512  010377  176114            MOV     R3,@RTCVT        ;RESTORE LINE CLOCK VECTOR
24483  126516  010477  176112            MOV     R4,@RTCPSW       ;RESTORE LINE CLOCK PSW VECTOR
24484
24485
24486                              ;******************************************************************************
  (2)                             ;TEST 630       TEST THAT RTC INTERRUPT CLEARS WITH RESET
  (3)                             ;******************************************************************************
  (2)  126522                   TS630:
24487  126522  032737  000001  001020        BIT     #1, @#$ENV       ;ARE WE RUNNING UNDER APT
  (1)  126530  001403                        BEQ     70$              ;IF NO THEN DO TEST
  (1)  126532  005737  001006               TST     @#$PASS          ;IS THIS FIRST PASS
  (2)  126536  001115                        BNE     TS631            ;IF NO THEN SHIP TO NEXT TEST
  (1)  126540                        70$:
24488  126540  106427  000340               MTPS    #340             ;SET PSW TO PRIORITY 7
24489  126544  017703  176062               MOV     @RTCVT,R3        ;SAVE LINE CLOCK VECTOR
24490  126550  012777  126614  176054       MOV     #2$,@RTCVT       ;POINT RTC VECTOR TO ERROR REPORT
24491  126556  005000                        CLR     R0
24492
24493  126560  052777  000100  176042       BIS     #BIT6,@LKS       ;ENABLE CLOCK INTERRUPTS
24494  126566  005200                   1$:  INC     R0
24495  126570  005700                        TST     R0
24496  126572  100375                        BPL     1$
24497  126574  000005                        RESET                    ;CLEAR PENDING INTERRUPT WITH RESET
24498  126576  106427  000240               MTPS    #240             ;SET PSW TO PRIORITY 5
24499  126602  000240                        NOP                      ;GIVE TIME FOR ANY INTERRUPT
24500  126604  042777  000100  176016       BIC     #BIT6,@LKS       ;DISALLOW INTERRUPTS
24501  126612  000401                        BR      3$
  (1)  126614                        2$:
  (2)  126614  104000                        EMT                      ;
24502  126616  010377  176010          3$:   MOV     R3,@RTCVT        ;RESTORE LINE CLOCK VECTOR
```

```
24503  126622  000446                          BR      SLU2ST
24504
24505
24506
24507  126624  004767  004600          ERROR6: JSR     PC,ABORT                ;ARE WE UNDER UFD ?
24508  126630  012737  000005  001002          MOV     #5,@#FATAL              ;SET UP FATAL ERROR NUMBER
24509  126636  012767  000001  052134          MOV     #1,$MSGTY               ;SET FATAL ERROR FLAG
24510  126644  032737  000001  001020          BIT     #1,@#ENV                ;UNDER APT /
24511  126652  001004                          BNE     LTCHLT
24512  126654  012700  126666                  MOV     #LTCMSG,R0
24513  126660  004767  004500                  JSR     PC,TYPE
24514  126664  000777                  LTCHLT: BR      .
24515  126666  040506  046111  051125  LTCMSG: .ASCIZ   /FAILURE DURING LTC TEST/<12><15>
       126674  020105  052504  044522
       126702  043516  046040  041524
       126710  052040  051505  005124
       126716  000015
24516                                   .EVEN
24517
24518                                   ;SERIAL LINE UNIT REGISTER AND VECTOR ADDRESSES FOR SLU2
24519
24520  126720  176500                  RCSR2:  176500                  ;ADDRESS OF RECEIVER COMMAND/STATUS REGISTER
24521  126722  176502                  RBUF2:  176502                  ;ADDRESS OF RECEIVER BUFFER
24522  126724  176504                  TCSR2:  176504                  ;ADDRESS OF TRANSMITTER COMMAND/STATUS REGISTER
24523  126726  176506                  TBUF2:  176506                  ;ADDRESS OF TRANSMITTER BUFFER
24524  126730  000300                  RVECT2: 300                     ;RECEIVER INTERRUPT VECTOR
24525  126732  000302                  RPSW2:  302
24526  126734  000304                  TVECT2: 304                     ;TRANSMITTER INTERRUPT VECTOR
24527  126736  000306                  TPSW2:  306
24528
24529  126740  000244                  SLU2ST: CLZ
24530  126742  032777  000020  072626          BIT     #20,@SWR
24531  126750  001402                          BEQ     1$
24532  126752  000167  002626                  JMP     UNIQUE
24533  126756  012737  000006  001004  1$:     MOV     #6,@#TESTN      ;PUT TEST NUMBER IN MAILBOX
24534  126764  012737  131474  000030          MOV     #ERROR7,@#30    ;SET UP FOR CORRECT ERROR CALL
24535
24536
24537                                   ;*****************************************************************************
  (2)                                   ;TEST 631       TEST ABILITY TO REFERENCE TCSR2
  (3)                                   ;*****************************************************************************
  (2)  126772                           TS631:
24538  126772  013703  000004                  MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
24539  126776  012737  127012  000004          MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
24540  127004  005777  177714                  TST     @TCSR2          ;REFERENCE THE XMIT COMMAND/STATUS REG.
24541  127010  000401                          BR      4$
  (1)  127012                           1$:
  (2)  127012  104000                          EMT                     ;
24542  127014  010337  000004          4$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
24543
24544
24545
24546                                   ;*****************************************************************************
  (2)                                   ;TEST 632       TEST ABILITY TO REFERENCE TBUF2
  (3)                                   ;*****************************************************************************
  (2)  127020                           TS632:
```

```
24547  127020  013703  000004                 MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
24548  127024  012737  127040  000004         MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
24549  127032  005777  177670                 TST     @TBUF2          ;REFERENCE THE XMIT BUFFER
24550  127036  000401                         BR      4$
  (1)  127040                          1$:
  (2)  127040  104000                         EMT                     ;
24551  127042  010337  000004          4$:    MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
24552
24553
24554                                ;*************************************************************************************
  (2)                                ;TEST 633       TEST THAT TCSR2 BIT7(DONE) CLEARS WHEN XBUF IS LOADED
  (3)                                ;*************************************************************************************
  (2)  127046                        TS633:
24555  127046  032737  000001  001020         BIT     #1, @#ENV       ;ARE WE RUNNING UNDER APT
  (1)  127054  001403                         BEQ     70$             ;IF NO THEN DO TEST
  (1)  127056  005737  001006                 TST     @#PASS          ;IS THIS FIRST PASS
  (2)  127062  001022                         BNE     TS634           ;IF NO THEN SHIP TO NEXT TEST
  (1)  127064                          70$:
24556  127064  005077  177636                 CLR     @TBUF2          ;LOAD XBUF
24557  127070  105777  177630                 TSTB    @TCSR2          ;CHECK DONE
24558  127074  100006                         BPL     3$              ;BR IF CLEAR
24559                                                                  ;FILL SECOND BUFFER BECUASE REFRESH COULD CAUSE
24560                                                                  ;FIRST TEST TO FAIL
24561  127076  005077  177624                 CLR     @TBUF2          ;FILL DOUBLE BUFFER
24562  127102  105777  177616                 TSTB    @TCSR2          ;CHECK DONE
24563  127106  100001                         BPL     3$
  (2)  127110  104000                         EMT                     ;
24564  127112  005000                  3$:    CLR     R0              ;CLEAR TIMER
24565  127114  105777  177604          4$:    TSTB    @TCSR2          ;CHECK FOR XMIT DONE
24566  127120  100403                         BMI     5$              ;IF DONE SETS, BR TO END OF TEST
24567  127122  005200                         INC     R0              ;INCREMENT TIMER
24568  127124  001373                         BNE     4$
  (2)  127126  104000                         EMT                     ;
24569  127130                           5$:
24570
24571
24572                                ;*************************************************************************************
  (2)                                ;TEST 634       TEST THAT TCSR2 "DONE" SETS WITH RESET
  (3)                                ;*************************************************************************************
  (2)  127130                        TS634:
24573  127130  032737  000001  001020         BIT     #1, @#ENV       ;ARE WE RUNNING UNDER APT
  (1)  127136  001403                         BEQ     70$             ;IF NO THEN DO TEST
  (1)  127140  005737  001006                 TST     @#PASS          ;IS THIS FIRST PASS
  (2)  127144  001015                         BNE     TS635           ;IF NO THEN SHIP TO NEXT TEST
  (1)  127146                          70$:
24574  127146  005077  177554                 CLR     @TBUF2          ;LOAD TRANSMIT BUFFER
24575  127152  105777  177546          1$:    TSTB    @TCSR2          ;WAIT FOR DONE
24576  127156  100375                         BPL     1$
24577  127160  005077  177542                 CLR     @TBUF2          ;LOAD SECOND BUFFER
24578  127164  000240                         NOP
24579  127166  000005                         RESET                   ;SET DONE WITH RESET
24580  127170  105777  177530                 TSTB    @TCSR2          ;CHECK FOR DONE SET
24581  127174  100401                         BMI     TS635
  (3)  127176  104000                         EMT                     ;
24582
24583
```

```
24584
24585                                   ;*****************************************************************************
  (2)                                   ;TEST 635       TEST ABILITY TO ACCESS RCSR2
  (3)                                   ;*****************************************************************************
  (2)  127200                           TS635:
24586  127200  013703  000004                  MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
24587  127204  012737  127220  000004          MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
24588  127212  005777  177502                  TST     @RCSR2          ;ACCESS RCSR
24589  127216  000401                          BR      2$
  (1)  127220                           1$:
  (2)  127220  104000                          EMT                     ;
24590  127222  010337  000004          2$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
24591
24592
24593                                   ;*****************************************************************************
  (2)                                   ;TEST 636       TEST ABILITY TO ACCESS RBUF2
  (3)                                   ;*****************************************************************************
  (2)  127226                           TS636:
24594  127226  013703  000004                  MOV     @#4,R3          ;SAVE TIMEOUT VECTOR
24595  127232  012737  127246  000004          MOV     #1$,@#4         ;SET UP TIMEOUT VECTOR
24596  127240  005777  177456                  TST     @RBUF2          ;ACCESS RBUF
24597  127244  000401                          BR      2$
  (1)  127246                           1$:
  (2)  127246  104000                          EMT                     ;
24598  127250  010337  000004          2$:     MOV     R3,@#4          ;RESTORE TIMEOUT VECTOR
24599
24600
24601
24602
24603
24604
24605                                   ;*****************************************************************************
  (2)                                   ;TEST 637       TEST THAT BIT0(BREAK BIT) CAN BE SET & CLEARED & RESET
  (3)                                   ;*****************************************************************************
  (2)  127254                           TS637:
24606  127254  032777  000001  177442          BIT     #BIT0,@TCSR2    ;CHECK BIT0 OF TCSR CLEAR
24607  127262  001401                          BEQ     3$
  (2)  127264  104000                          EMT                     ;
24608  127266  052777  000001  177430  3$:     BIS     #BIT0,@TCSR2    ;SET BIT0 IN TCSR
24609  127274  032777  000001  177422          BIT     #BIT0,@TCSR2    ;TEST BIT0 OF TCSR
24610  127302  001001                          BNE     4$
  (2)  127304  104000                          EMT                     ;
24611  127306  042777  000001  177410  4$:     BIC     #BIT0,@TCSR2    ;CLEAR BIT0 OF TCSR
24612  127314  032777  000001  177402          BIT     #BIT0,@TCSR2    ;TEST BIT0 OF TCSR
24613  127322  001401                          BEQ     7$
  (2)  127324  104000                          EMT                     ;
24614  127326                           7$:
  (1)  127326  032737  000001  001020          BIT     #1, @#$ENV      ;ARE WE RUNNING UNDER APT
  (1)  127334  001403                          BEQ     70$             ;IF NO THEN DO TEST
  (1)  127336  005737  001006                  TST     @#$PASS         ;IS THIS FIRST PASS
  (2)  127342  001011                          BNE     TS640           ;IF NO THEN SHIP TO NEXT TEST
  (1)  127344                           70$:
24615  127344  052777  000001  177352          BIS     #BIT0,@TCSR2    ;SET BIT0 IN TCSR
24616  127352  000005                          RESET                   ;CLEAR BIT0 WITH RESET
24617  127354  032777  000001  177342          BIT     #BIT0,@TCSR2    ;TEST BIT0 CLEAR
24618  127362  001401                          BEQ     TS640
```

C3



```
  (3)  127364  104000                          EMT                     ;
24619
24620
24621                                  ;*****************************************************************************
  (2)                                  ;TEST 640        TEST THAT BIT6(XMIT INT EN) CAN BE SET & RESET
  (3)                                  ;*****************************************************************************
  (2)  127366                          TS640:
24622  127366  017703  177342                  MOV     @TVECT2,R3      ;SAVE XMIT VECTOR
24623  127372  012777  127414  177334          MOV     #1$,@TVECT2     ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
24624  127400  106427  000340                  MTPS    #340            ;SET PSW TO PRIORITY 7
24625  127404  032777  000100  177312          BIT     #BIT6,@TCSR2    ;TEST BIT6 OF TCSR
24626  127412  001401                          BEQ     2$
  (1)  127414                          1$:
  (2)  127414  104000                          EMT                     ;
24627  127416  052777  000100  177300  2$:     BIS     #BIT6,@TCSR2    ;SET BIT6 OF TCSR
24628  127424  032777  000100  177272          BIT     #BIT6,@TCSR2    ;TEST BIT6 OF TCSR
24629  127432  001001                          BNE     3$
  (2)  127434  104000                          EMT                     ;
24630  127436  042777  000100  177260  3$:     BIC     #BIT6,@TCSR2    ;CLEAR BIT6 OF TCSR
24631  127444  032777  000100  177252          BIT     #BIT6,@TCSR2    ;TEST BIT6 OF TCSR
24632  127452  001401                          BEQ     4$
  (2)  127454  104000                          EMT                     ;
24633  127456  032737  000001  001020  4$:     BIT     #1,@#$ENV       ;ARE WE RUNNING UNDER APT
24634  127464  001403                          BEQ     70$             ;IF NO THEN DO TEST
24635  127466  005737  001006                  TST     @#$PASS         ;IF THIS FIRST PASS
24636  127472  001011                          BNE     5$              ;IF NO THEN SKIP TO END OF TEST
24637  127474                          70$:
24638  127474  052777  000100  177222          BIS     #BIT6,@TCSR2    ;SET BIT6 OF TCSR
24639  127502  000005                          RESET                   ;CLEAR BIT6 WITH RESET
24640  127504  032777  000100  177212          BIT     #BIT6,@TCSR2    ;TEST BIT6 OF TCSR
24641  127512  001401                          BEQ     5$
  (2)  127514  104000                          EMT                     ;
24642  127516  010377  177212          5$:     MOV     R3,@TVECT2      ;RESTORE XMIT VECTOR
24643
24644
24645                                  ;*****************************************************************************
  (2)                                  ;TEST 641        TEST THAT BIT6 OF RCSR2 CAN BE SET & RESET
  (3)                                  ;*****************************************************************************
  (2)  127522                          TS641:
24646  127522  017703  177202                  MOV     @RVECT2,R3      ;SAVE RECEIVE VECTOR
24647  127526  012777  127550  177174          MOV     #1$,@RVECT2     ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
24648  127534  106427  000340                  MTPS    #340            ;SET PSW TO PRIORITY 7
24649  127540  032777  000100  177152          BIT     #BIT6,@RCSR2    ;TEST BIT6 OF RCSR
24650  127546  001401                          BEQ     2$
  (1)  127550                          1$:
  (2)  127550  104000                          EMT                     ;
24651  127552  052777  000100  177140  2$:     BIS     #BIT6,@RCSR2    ;SET BIT6 OF RCSR
24652  127560  032777  000100  177132          BIT     #BIT6,@RCSR2    ;TEST BIT6 OF RCSR
24653  127566  001001                          BNE     3$
  (2)  127570  104000                          EMT                     ;
24654  127572  042777  000100  177120  3$:     BIC     #BIT6,@RCSR2    ;CLEAR BIT6 OF RCSR
24655  127600  032777  000100  177112          BIT     #BIT6,@RCSR2    ;TEST BIT6 OF RCSR
24656  127606  001401                          BEQ     4$
  (2)  127610  104000                          EMT                     ;
24657  127612  032737  000001  001020  4$:     BIT     #1,@#$ENV       ;ARE WE RUNNING UNDER APT
24658  127620  001403                          BEQ     70$             ;IF NO THEN DO TEST
```

```
127622  005737  001006                  TST     @#$PASS          ;IS THIS FIRST PASS
127626  001011                          BNE     5$               ;IF NO THEN SKIP TO END OF TEST
127630                          70$:
127630  052777  000100  177062          BIS     #BIT6,@RCSR2     ;SET BIT6 OF RCSR
127636  000005                          RESET                    ;CLEAR BIT6OF RCSR2 WITH RESET
127640  032777  000100  177052          BIT     #BIT6,@RCSR2     ;TEST BIT6 OF RCSR
127646  001401                          BEQ     5$
127650  104000                          EMT                      ;
127652  010377  177052          5$:     MOV     R3,@RVECT2       ;RESTORE RECEIVE VECTOR



                                ;*******************************************************************************
                                ;TEST 642       TEST THAT XMIT INTERRUPTS ONLY WHEN ENABLED
                                ;*******************************************************************************
127656                          TS642:
127656  042777  000100  177040          BIC     #BIT6,@TCSR2     ;CLEAR TRANSMIT INTERRUPT ENABLE
127664  017703  177044                  MOV     @TVECT2,R3       ;SAVE XMIT VECTOR
127670  012777  127712  177036          MOV     #2$,@TVECT2      ;POINT XMIT VECTOR TO ERROR REPORT
127676  105777  177022          1$:     TSTB    @TCSR2           ;WAIT FOR DONE
127702  100375                          BPL     1$
127704  106427  000140                  MTPS    #140             ;SET PSW TO PRIORITY 3
127710  000401                          BR      3$
127712                          2$:
127712  104000                          EMT                      ;
127714  012777  127734  177012  3$:     MOV     #4$,@TVECT2      ;SET XMIT VECTOR TO END OF TEST
127722  052777  000100  176774          BIS     #BIT6,@TCSR2     ;ENABLE INTERRUPTS
127730  000240                          NOP

127732  104000                          EMT                      ;XMIT DID NOT INTERRUPT

127734  042777  000100  176762  4$:     BIC     #BIT6,@TCSR2     ;DISABLE INTERRUPTS
127742  022626                          CMP     (SP)+,(SP)+      ;RESTORE SP AFTER INTERRUPT
127744  010377  176764                  MOV     R3,@TVECT2       ;RESTORE XMIT VECTOR



                                ;*******************************************************************************
                                ;TEST 643       TEST THAT XMIT INTERRUPTS DO NOT OCCUR WHEN DISABLED
                                ;*******************************************************************************
127750                          TS643:
127750  042777  000100  176746          BIC     #BIT6,@TCSR2     ;DISABLE INTERRUPTS
127756  106427  000340                  MTPS    #340             ;SET PSW TO PRIORITY 7
127762  017703  176746                  MOV     @TVECT2,R3       ;SAVE XMIT VECTOR
127766  012777  130014  176740          MOV     #2$,@TVECT2      ;POINT XMIT VECTOR TO ERROR REPORT
127774  105777  176724          1$:     TSTB    @TCSR2           ;WAIT FOR DONE
130000  100375                          BPL     1$
130002  052777  000100  176714          BIS     #BIT6,@TCSR2     ;ENABLE INTERRUPT
130010  000240                          NOP
130012  000401                          BR      3$
130014                          2$:
130014  104000                          EMT                      ;
130016  042777  000100  176700  3$:     BIC     #BIT6,@TCSR2     ;CLEAR INTERRUPT ENABLE
130024  012777  130042  176702          MOV     #4$,@TVECT2      ;POINT XMIT VECTOR TO ERROR REPORT
130032  106427  000140                  MTPS    #140             ;SET PSW TO PRIORITY 3
130036  000240                          NOP
130040  000401                          BR      5$
```

```
  (1)  130042                                  4$:
  (2)  130042  104000                                  EMT                          ;
24704  130044  010377  176664                  5$:     MOV     R3,@TVECT2           ;RESTORE XMIT VECTOR
24705
24706
24707                                          ;*******************************************************************************
  (2)                                          ;TEST 644       TEST TRANSMITTER FOR DOUBLE INTERRUPTS
  (3)                                          ;*******************************************************************************
  (2)  130050                                  TS644:
24708  130050  042777  000100  176646                  BIC     #BIT6,@TCSR2         ;CLEAR INTERRUPT ENABLE
24709  130056  017703  176652                          MOV     @TVECT2,R3           ;SAVE XMIT VECTOR
24710  130062  017704  176650                          MOV     @TPSW2,R4            ;SAVE XMIT PSW VECTOR
24711  130066  012777  130126  176640                  MOV     #2$,@TVECT2          ;SET UP XMIT VECTOR
24712  130074  012777  000340  176634                  MOV     #340,@TPSW2          ;SET PIO 7 AFTER INTERRUPT
24713  130102  106427  000140                          MTPS    #140                 ;SET PSW TO PRIORITY 3
24714  130106  105777  176612                  1$:     TSTB    @TCSR2               ;WAIT FOR DONE
24715  130112  100375                                  BPL     1$
24716  130114  052777  000100  176602                  BIS     #BIT6,@TCSR2         ;ENABLE INTERRUPTS
24717  130122  000240                                  NOP
24718
24719  130124  104000                                  EMT                          ;
24720                                                                               ;XMIT INTERRUPT DID NOT OCCUR
24721  130126  022626                          2$:     CMP     (SP)+,(SP)+          ;RESTORE SP AFTER INTERRUPT
24722  130130  012777  130154  176576                  MOV     #4$,@TVECT2          ;POINT XMIT VECTOR TO ERROR
24723  130136  106427  000140                          MTPS    #140                 ;SET PSW TO PRIORITY 3
24724  130142  000240                                  NOP                          ;GIVE TIME FOR ANY INTERRUPTS
24725  130144  042777  000100  176552                  BIC     #BIT6,@TCSR2         ;DISABLE INTERRUPTS
24726  130152  000401                                  BR      5$
  (1)  130154                                  4$:
  (2)  130154  104000                                  EMT                          ;
24727  130156  010377  176552                  5$:     MOV     R3,@TVECT2           ;RESTORE XMIT VECTOR
24728  130162  010477  176550                          MOV     R4,@TPSW2            ;RESTORE XMIT PSW VECTOR
24729
24730                                          ;*******************************************************************************
  (2)                                          ;TEST 645       TEST THAT XMIT INTERRUPT CLEARS WITH LOADING TBUF2
  (3)                                          ;*******************************************************************************
  (2)  130166                                  TS645:
24731  130166  032737  000001  001020                  BIT     #1, @#$ENV           ;ARE WE RUNNING UNDER APT
  (1)  130174  001403                                  BEQ     70$                  ;IF NO THEN DO TEST
  (1)  130176  005737  001006                          TST     @#$PASS              ;IS THIS FIRST PASS
  (2)  130202  001043                                  BNE     TS646                ;IF NO THEN SHIP TO NEXT TEST
  (1)  130204                                  70$:
24732  130204  042777  000100  176512                  BIC     #BIT6,@TCSR2         ;DISABLE INTERRUPTS
24733  130212  106427  000340                          MTPS    #340                 ;SET PSW TO PRIORITY 7
24734  130216  017703  176512                          MOV     @TVECT2,R3           ;SAVE XMIT VECTOR
24735  130222  012777  130272  176504                  MOV     #2$,@TVECT2          ;POINT XMIT VECTOR TO ERROR
24736  130230  052777  000100  176466                  BIS     #BIT6,@TCSR2         ;ENABLE INTERRUPTS
24737  130236  005077  176464                          CLR     @TBUF2               ;LOAD TBUF
24738  130242  105777  176456                  1$:     TSTB    @TCSR2               ;WAIT FOR DONE (INTERRUPT)
24739  130246  100375                                  BPL     1$
24740  130250  005077  176452                          CLR     @TBUF2               ;FILL SECOND BUFFER TO RESET INT.
24741  130254  106427  000140                          MTPS    #140                 ;SET PSW TO PRIORITY 3
24742  130260  000240                                  NOP                          ;GIVE TIME FOR ANY INTERRUPTS
24743  130262  042777  000100  176434                  BIC     #BIT6,@TCSR2         ;DISABLE INTERRUPTS
24744  130270  000401                                  BR      3$
  (1)  130272                                  2$:
```

F3



```
  (2)  130272  104000                          EMT                     ;
24745  130274  010377  176434          3$:     MOV     R3,@TVECT2      ;RESTORE XMIT VECTOR
24746  130300  005000                          CLR     R0              ;INIT LOOP COUNTER
24747  130302  005200                  4$:     INC     R0              ;INCREMENT COUNTER
24748  130304  001376                          BNE     4$              ;UNTIL COUNTER = 0
24749  130306  005777  176410                  TST     @RBUF2          ;CLEAR RECEIVER BUFFER
24750
24751
24752                                  ;*******************************************************************************
  (2)                                  ;TEST 646       TEST THAT RCVR DONE (7) SET & CLEAR PROPERLY
  (3)                                  ;*******************************************************************************
  (2)  130312                          TS646:
24753  130312  005000                          CLR     R0              ;CLEAR A TIMER
24754  130314  005077  176406                  CLR     @TBUF2          ;LOAD TRANSMIT BUFFER
24755  130320  105777  176374          WDONE2: TSTB    @RCSR2          ;CHECK FOR RECEIVER DONE
24756  130324  100403                          BMI     6$              ;BR, IF DONE
24757  130326  005200                          INC     R0              ;INCREMENT TIMER, IF NOT DONE
24758  130330  001373                          BNE     WDONE2
  (2)  130332  104000                          EMT                     ;RECEIVER DONE NEVER SET
24759
24760  130334  032737  000001  001020  6$:     BIT     #1,@#$ENV       ;ARE WE RUNNING UNDER APT
24761  130342  001403                          BEQ     70$             ;IF NO THEN DO TEST
24762  130344  005737  001006                  TST     @#$PASS         ;IS THIS FIRST PASS
24763  130350  001005                          BNE     2$              ;IF NO THEN SKIP TO END OF TEST
24764  130352                          70$:
24765  130352  000005                          RESET                   ;CLEAR DONE WITH RESET
24766  130354  105777  176340                  TSTB    @RCSR2          ;CHECK FOR DONE CLEAR
24767  130360  001401                          BEQ     2$
  (2)  130362  104000                          EMT                     ;RESET DID NOT CLEAR RCVR DONE
24768  130364  005000                  2$:     CLR     R0              ;INIT LOOP COUNTER
24769  130366  005200                  3$:     INC     R0              ;INCREMENT COUNTER
24770  130370  001376                          BNE     3$              ;UNTIL COUNTER = 0
24771  130372  005777  176324                  TST     @RBUF2          ;CLEAR RECEIVER BUFFER
24772
24773
24774                                  ;*******************************************************************************
  (2)                                  ;TEST 647       TEST THAT READING RBUF2 CLEARS RECEIVER DONE
  (3)                                  ;*******************************************************************************
  (2)  130376                          TS647:
24775  130376  005077  176324                  CLR     @TBUF2          ;LOAD TRANSMITTER
24776  130402  105777  176312          1$:     TSTB    @RCSR2          ;WAIT FOR RECEIVER DONE
24777  130406  100375                          BPL     1$
24778  130410  017700  176306                  MOV     @RBUF2,R0       ;READ RECEIVE BUFFER
24779  130414  105777  176300                  TSTB    @RCSR2          ;CHECK FOR RECEIVE DONE CLEAR
24780  130420  001401                          BEQ     TS650
  (3)  130422  104000                          EMT                     ;
24781                                                                  ;READING RBUF2 DID NOT CLEAR RCVR DONE
24782
24783
24784                                  ;*******************************************************************************
  (2)                                  ;TEST 650       TEST THAT RCVR INTERRUPTS ONLY WHEN ENABLED
  (3)                                  ;*******************************************************************************
  (2)  130424                          TS650:
24785  130424  042777  000100  176272          BIC     #BIT6,@TCSR2    ;DISABLE TRANSMIT INTERRUPTS
24786  130432  042777  000100  176260          BIC     #BIT6,@RCSR2    ;DISABLE RECEIVER INTERRUPTS
24787  130440  017703  176264                  MOV     @RVECT2,R3      ;SAVE RECEIVE VECTOR
```

```
24788  130444  012777  130472  176256          MOV     #2$,@RVECT2     ;POINT RCV VECTOR TO ERROR REPORT
24789  130452  106427  000140                  MTPS    #140            ;SET PSW TO PRIORITY 3
24790  130456  005077  176244                  CLR     @TBUF2          ;SEND A CHARACTER
24791  130462  105777  176232          1$:     TSTB    @RCSR2          ;WAIT FOR RECEIVER DONE
24792  130466  100375                          BPL     1$
24793  130470  000401                          BR      3$
  (1)  130472                          2$:
  (2)  130472  104000                          EMT                     ;
24794  130474  012777  130514  176226  3$:     MOV     #4$,@RVECT2     ;POINT RCV VECTOR TO END OF TEST
24795  130502  052777  000100  176210          BIS     #BIT6,@RCSR2    ;ENABLE RCV INTERRUPTS
24796  130510  000240                          NOP                     ;GIVE ANY INTERRUPTS TIME
24797  130512  104000                          EMT                     ;
24798  130514  042777  000100  176176  4$:     BIC     #BIT6,@RCSR2    ;DISABLE INTERRUPTS
24799  130522  022626                          CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
24800  130524  005777  176172                  TST     @RBUF2          ;CLEAR CHARACTER FROM RECEIVER BUFFER
24801  130530  010377  176174                  MOV     R3,@RVECT2      ;RESTORE RECEIVE VECTOR
24802
24803
24804                                ;*************************************************************************
  (2)                              ;TEST 651       TEST THAT RCVR INTERRUPTS DO NOT OCCUR WHEN DISABLED
  (3)                              ;*************************************************************************
  (2)  130534                      TS651:
24805  130534  106427  000340                  MTPS    #340            ;SET PSW TO PRIORITY 7
24806  130540  017703  176164                  MOV     @RVECT2,R3      ;SAVE RECEIVE VECTOR
24807  130544  012777  130576  176156          MOV     #2$,@RVECT2     ;POINT RCVR VECTOR TO ERROR REPORT
24808  130552  005077  176150                  CLR     @TBUF2          ;SEND A CHARACTER
24809  130556  105777  176136          1$:     TSTB    @RCSR2          ;WAIT FOR RECEIVER DONE
24810  130562  100375                          BPL     1$
24811  130564  052777  000100  176126          BIS     #BIT6,@RCSR2    ;ENABLE INTERRUPTS
24812  130572  000240                          NOP                     ;GIVE TIME FOR INTERRUPT
24813  130574  000401                          BR      3$
  (1)  130576                          2$:
  (2)  130576  104000                          EMT                     ;RCVR INTERRUPTS AT PRIORITY 7
24814  130600  042777  000100  176112  3$:     BIC     #BIT6,@RCSR2    ;CLEAR INTERRUPT ENABLE
24815  130606  012777  130624  176114          MOV     #4$,@RVECT2     ;POINT RCVR VECTOR TO ERROR REPORT
24816  130614  106427  000140                  MTPS    #140            ;SET PSW TO PRIORITY 3
24817  130620  000240                          NOP                     ;GIVE TIME FOR ANY INTERRUPT
24818  130622  000401                          BR      5$
  (1)  130624                          4$:
  (2)  130624  104000                          EMT                     ;RCVR INTERRUPT REQUEST PASSED WITH BIT6 CLEAR
24819  130626  005777  176070          5$:     TST     @RBUF2          ;CLEAR CHARACTER FROM RECEIVER BUFFER
24820  130632  010377  176072                  MOV     R3,@RVECT2      ;RESTORE RECEIVE VECTOR
24821
24822
24823                                ;*************************************************************************
  (2)                              ;TEST 652       TEST RECEIVER FOR DOUBLE INTERRUPTS
  (3)                              ;*************************************************************************
  (2)  130636                      TS652:
24824  130636  017703  176066                  MOV     @RVECT2,R3      ;SAVE RECEIVE VECTOR
24825  130642  017704  176064                  MOV     @RPSW2,R4       ;SAVE RECEIVE PSW VECTOR
24826  130646  012777  130712  176054          MOV     #2$,@RVECT2     ;POINT RCV VECTOR TO CONTINUE TEST
24827  130654  012777  000340  176050          MOV     #340,@RPSW2     ;SET PRIORITY TO 7 AFTER INTERRUPT
24828  130662  106427  000140                  MTPS    #140            ;SET PSW TO PRIORITY 3
24829  130666  005077  176034                  CLR     @TBUF2          ;SEND A CHARACTER
24830  130672  105777  176022          1$:     TSTB    @RCSR2          ;WAIT FOR RCVR DONE
24831  130676  100375                          BPL     1$
```

```
24832  130700  052777  000100  176012          BIS     #BIT6,@RCSR2    ;ENABLE RCV INTERRUPTS
24833  130706  000240                          NOP                     ;GIVE SOME TIME
24834  130710  104000                          EMT                     ;
24835  130712  022626                  2$:     CMP     (SP)+,(SP)+     ;RESTORE SP AFTER INTERRUPT
24836  130714  012777  130750  176006          MOV     #3$,@RVECT2     ;POINT RCV VECTOR TO ERROR REPORT
24837  130722  106427  000140                  MTPS    #140            ;SET PSW TO PRIORITY 3
24838  130726  000240                          NOP                     ;GIVE SOME TIME
24839  130730  042777  000100  175762          BIC     #BIT6,@RCSR2    ;CLEAR INTERRUPT ENABLE
24840  130736  010377  175760                  MOV     R3,@RVECT2      ;RESTORE RECEIVE VECTOR
24841  130742  010477  175764                  MOV     R4,@RPSW2       ;RESTORE RECEIVE PSW VECTOR
24842  130746  000401                          BR      4$
  (1)  130750                          3$:
  (2)  130750  104000                          EMT                     ;
24843  130752  005777  175744          4$:     TST     @RBUF2          ;CLEAR CHARACTER FROM RECEIVER BUFFER
24844  130756  010377  175746                  MOV     R3,@RVECT2      ;RESTORE RECEIVE VECTOR
24845
24846
24847                                  ;*****************************************************************************
  (2)                                  ;TEST 653       TEST THAT RCVR INTERRUPT CLEARS BY READING RBUF2
  (3)                                  ;*****************************************************************************
  (2)  130762                          TS653:
24848  130762  106427  000340                  MTPS    #340            ;SET PSW TO PRIORITY 7
24849  130766  017703  175736                  MOV     @RVECT2,R3      ;SAVE RECEIVE VECTOR
24850  130772  012777  131042  175730          MOV     #2$,@RVECT2     ;POINT RCV VECTOR TO ERROR REPORT
24851  131000  052777  000100  175712          BIS     #BIT6,@RCSR2    ;SET RCVR INTERRUPT ENABLE
24852  131006  005077  175714                  CLR     @TBUF2          ;SEND A CHARACTER
24853  131012  105777  175702          1$:     TSTB    @RCSR2          ;WAIT FOR DONE (INTERRUPT)
24854  131016  100375                          BPL     1$
24855  131020  005777  175676                  TST     @RBUF2          ;READ RBUF TO CLEAR PENDING INTERRPUT
24856  131024  106427  000140                  MTPS    #140            ;SET PSW TO PRIORITY 3
24857  131030  000240                          NOP                     ;ALLOW TIME FOR ANY ERRONEOUS INTERRUPT
24858  131032  042777  000100  175660          BIC     #BIT6,@RCSR2    ;NO INTERRUPT-CLEAR INT. ENABLE
24859  131040  000401                          BR      3$
  (1)  131042                          2$:
  (2)  131042  104000                          EMT                     ;
24860  131044  010377  175660          3$:     MOV     R3,@RVECT2      ;RESTORE RECEIVE VECTOR
24861
24862
24863
24864                                  ;*****************************************************************************
  (2)                                  ;TEST 654       TEST THAT RESET CLEARS RECEIVE INTERRUPT
  (3)                                  ;*****************************************************************************
  (2)  131050                          TS654:
24865  131050  032737  000001  001020          BIT     #1, @#$ENV      ;ARE WE RUNNING UNDER APT
  (1)  131056  001403                          BEQ     70$             ;IF NO THEN DO TEST
  (1)  131060  005737  001006                  TST     @#$PASS         ;IS THIS FIRST PASS
  (2)  131064  001036                          BNE     TS655           ;IF NO THEN SHIP TO NEXT TEST
  (1)  131066                          70$:
24866  131066  106427  000340                  MTPS    #340            ;SET PSW TO PRIORITY 7
24867  131072  017703  175632                  MOV     @RVECT2,R3      ;SAVE RECEIVE VECTOR
24868  131076  012777  131154  175624          MOV     #2$,@RVECT2     ;POINT RCV VECTOR TO ERROR REPORT
24869  131104  052777  000100  175606          BIS     #BIT6,@RCSR2    ;SET RCV INTERRUPT ENABLE
24870  131112  012777  000377  175606          MOV     #377,@TBUF2     ;SEND AN ALL 1'S CHARACTER
24871  131120  105777  175574          1$:     TSTB    @RCSR2          ;WAIT FOR RCV DONE
24872  131124  100375                          BPL     1$
24873  131126  000005                          RESET                   ;CLEAR RCV INTERRUPT & RBUF2
```

```
24874  131130  052777  000100  175562          BIS     #BIT6,@RCSR2    ;SET RECEIVER INTERRUPT
24875  131136  106427  000140                  MTPS    #140            ;SET PSW TO PRIORITY 3
24876  131142  000240                          NOP                     ;ALLOW TIME FOR AN ERRONEOUS INTERRUPT
24877  131144  042777  000100  175546          BIC     #BIT6,@RCSR2    ;NO INTERRUPT-CLEAR INT. ENABLE
24878  131152  000401                          BR      3$
  (1)  131154                          2$:
  (2)  131154  104000                          EMT                     ;
24879  131156  010377  175546          3$:     MOV     R3,@RVECT2      ;RESTORE RECEIVE VECTOR
24880
24881
24882                                ;*************************************************************************************
  (2)                               ;TEST 655       TEST THAT THE "OR" ERROR (BIT14) & "ERROR" (BIT15) CAN BE SET
  (3)                               ;*************************************************************************************
  (2)  131162                       TS655:
24883  131162  012700  000003                  MOV     #3,R0           ;SET CHARACTER COUNT TO SEND 3 CHAR.
24884  131166  005077  175534          1$:     CLR     @TBUF2          ;LOAD TRANSMIT BUFFER
24885  131172  105777  175526          2$:     TSTB    @TCSR2          ;WAIT FOR TRANSMIT DONE
24886  131176  100375                          BPL     2$
24887  131200  005300                          DEC     R0              ;DECREMENT CHARACTER COUNT
24888  131202  001371                          BNE     1$              ;BR IF ALL CHARACTERS NOT TRANSMITTED
24889  131204  032777  040000  175510          BIT     #BIT14,@RBUF2   ;TEST FOR "OR" ERROR FLAG
24890  131212  001001                          BNE     3$
  (2)  131214  104000                          EMT                     ;
24891  131216  032777  100000  175476  3$:     BIT     #BIT15,@RBUF2   ;TEST "ERROR" FLAG
24892  131224  001001                          BNE     4$
  (2)  131226  104000                          EMT                     ;
24893  131230  005000                  4$:     CLR     R0              ;CLEAR LOOP COUNTER
24894  131232  005200                  5$:     INC     R0              ;INCREMENT LOOP COUNTER
24895  131234  001376                          BNE     5$              ;  UNTIL COUNTER = 0
24896  131236  005777  175460                  TST     @RBUF2          ;CLEAR CHARACTER FROM RECEIVER BUFFER
24897
24898
24899                                ;*************************************************************************************
  (2)                               ;TEST 656       TEST THAT BREAK TRANSMITS ALL ZEROES
  (3)                               ;*************************************************************************************
  (2)  131242                       TS656:
24900  131242  032737  000001  001020          BIT     #1, @#$ENV      ;ARE WE RUNNING UNDER APT
  (1)  131250  001403                          BEQ     70$             ;IF NO THEN DO TEST
  (1)  131252  005737  001006                  TST     @#$PASS         ;IS THIS FIRST PASS
  (2)  131256  001027                          BNE     TS657           ;IF NO THEN SHIP TO NEXT TEST
  (1)  131260                       70$:
24901  131260  012777  177777  175440          MOV     #-1,@TBUF2      ;TRANSMIT ALL ONES TO RCVR
24902  131266  105777  175426          1$:     TSTB    @RCSR2          ;WAIT FOR RCVR DONE
24903  131272  100375                          BPL     1$
24904  131274  005777  175422                  TST     @RBUF2          ;CLEAR DONE (LEAVING ALL ONES IN RBUF)
24905  131300  052777  000001  175416          BIS     #BIT0,@TCSR2    ;TRANSMIT BREAK
24906  131306  005000                          CLR     R0              ;CLEAR A TIMER
24907  131310  105777  175404          2$:     TSTB    @RCSR2          ;WAIT FOR RCVR DONE
24908  131314  100403                          BMI     CONT42          ;BR IF DONE
24909  131316  005200                          INC     R0              ;IF NOT, INCREMENT TIMER
24910  131320  001373                          BNE     2$
  (2)  131322  104000                          EMT                     ;BREAK DID NOT TRANSMIT ANYTHING
24911
24912  131324  105777  175372          CONT42: TSTB    @RBUF2          ;CHECK RECEIVE BUFFER FOR ZERO
24913  131330  001401                          BEQ     3$
  (2)  131332  104000                          EMT                     ;BREAK DID NOT TRANSMIT ALL ZEROES
```

```
24914
24915  131334  000005                  3$:     RESET                   ;CLEAR ERRORS
24916
24917
24918                                  ;*****************************************************************************
  (2)                                  ;TEST 657       TEST THAT "FR" ERROR CAN BE SET DURING BREAK
  (3)                                  ;*****************************************************************************
  (2)  131336                          TS657:
24919  131336  052777  000001  175360          BIS     #BIT0,@TCSR2    ;SEND BREAK
24920  131344  005077  175356          CLR     @TBUF2          ;TRANSMIT A CHARACTER TO TIME BREAK
24921  131350  105777  175344  1$:     TSTB    @RCSR2          ;WAIT FOR RCVR DONE
24922  131354  100375                  BPL     1$
24923  131356  042777  000001  175340  BIC     #BIT0,@TCSR2    ;CLEAR BREAK BITS
24924  131364  032777  020000  175330  BIT     #BIT13,@RBUF2   ;CHECK FOR FRAMING ERROR FLAG
24925  131372  001001                  BNE     2$
  (2)  131374  104000                  EMT                     ;BREAK DID NOT SET FRAMING ERROR
24926  131376  032777  100000  175316  2$:     BIT     #BIT15,@RBUF2   ;TEST "ERROR"FLAG
24927  131404  001001                  BNE     3$
  (2)  131406  104000                  EMT                     ;"ERROR" FLAG DID NOT SET WITH "OR" FLAG
24928  131410  005777  175306          3$:     TST     @RBUF2          ;CLEAR RECEIVER BUFFER
24929
24930                                  ;*****************************************************************************
  (2)                                  ;TEST 660       TEST DATA PATHS USING WRAP CABLE
  (3)                                  ;*****************************************************************************
  (2)  131414                          TS660:
24931  131414  005001                  CLR     R1              ;CLEAR REGISTER FOR TEST DATA
24932  131416  105201          1$:     INCB    R1              ;INCREMENT THE TEST DATA
24933  131420  010177  175302          MOV     R1,@TBUF2       ;XMIT A CHARACTER
24934  131424  005000                  CLR     R0              ;CLEAR A TIMER
24935  131426  105777  175266  2$:     TSTB    @RCSR2          ;WAIT FOR RECEIVER DONE
24936  131432  100403                  BMI     3$              ;BR IF DONE
24937  131434  005200                  INC     R0              ;INCREMENT TIMER IF NOT
24938  131436  001373                  BNE     2$
  (2)  131440  104000                  EMT                     ;
24939  131442  017702  175254  3$:     MOV     @RBUF2,R2       ;GET RECEIVED CHARACTER
24940  131446  020102                  CMP     R1,R2           ;COMPARE DATA
24941  131450  001401                  BEQ     4$
  (2)  131452  104000                  EMT                     ;
24942  131454  105701          4$:     TSTB    R1              ;TEST XMIT DATA FOR ZERO
24943  131456  001357                  BNE     1$              ;BR, IF NOT FINISHED
24944  131460  000167  000120          JMP     UNIQUE          ;FINISHED TESTING DEVICES SEPARATELY
24945                                                          ;GO TEST THEM ALL TOGETHER
24946  131464  000000          $BDADR: 0
24947  131466  000000          $BDDAT: 0
24948  131470  000000          $GDADR: 0
24949  131472  000000          $GDDAT: 0
24950
24951
24952  131474  004767  001730  ERROR7: JSR     PC,ABORT                ;ARE WE UNDER UFD ?
24953  131500  012737  000006  001002  MOV     #6,@#$FATAL             ;SET UP FATAL ERROR NUMBER
24954  131506  012767  000001  047264  MOV     #1,$MSGTY               ;SET FATAL ERROR FLAG
24955  131514  032737  000001  001020  BIT     #1,@#$ENV               ;UNDER APT ?
24956  131522  001004                  BNE     SL2HLT                  ;YES
24957  131524  012700  131536          MOV     #SL2MSG,R0
24958  131530  004767  001630          JSR     PC,TYPE
24959  131534  000777          SL2HLT: BR      .
```

```
24960
24961  131536 040506 046111 051125  SL2MSG: .ASCIZ  /FAILURE DURING SLU 2 TEST/<12><15>
       131544 020105 052504 044522
       131552 043516 051440 052514
       131560 031040 052040 051505
       131566 005124 000015
24962                               .EVEN
24963
24964
24965  131572 177560                DADTBL: .WORD   177560
24966  131574 177564                        .WORD   177564
24967  131576 176500                        .WORD   176500
24968  131600 176504                        .WORD   176504
24969  131602 177564                TBLEND: .WORD   177564
24970
24971
24972  131604 032737 000040 000052  UNIQUE: BIT     #40,@#52        ;ARE WE RUNNING UNDER UFD ?
24973  131612 001002                        BNE     10$             ;NO,THEN CONTINUE
24974  131614 000167 001340                 JMP     ENDPAS          ;YES,THEN SKIP TESTS
24975  131620 032777 000034 067750  10$:    BIT     #34,@SWR
24976  131626 001402                        BEQ     1$
24977  131630 000167 001324                 JMP     ENDPAS
24978  131634 012737 000007 001004  1$:     MOV     #7,@#TESTN      ;UPDATE TEST NUMBER FOR APT
24979  131642 012737 132460 000030          MOV     #ERROR8,@#30    ;SET UP FOR CORRECT ERROR CALL
24980
24981                               ;*****************************************************************************
  (2)                               ;TEST 661       UNIQUE INTERNAL ADDRESS TEST
  (3)                               ;*****************************************************************************
  (2)  131650                       TS661:
24982  131650 032737 000001 001020          BIT     #1, @#ENV       ;ARE WE RUNNING UNDER APT
  (1)  131656 001403                         BEQ     70$             ;IF NO THEN DO TEST
  (1)  131660 005737 001006                  TST     @#PASS          ;IS THIS FIRST PASS
  (2)  131664 001044                         BNE     TS662           ;IF NO THEN SHIP TO NEXT TEST
  (1)  131666                       70$:
24983  131666 012767 000340 046102          MOV     #340,PS         ;WE WILL BE PLAYING WITH BIT6
24984                                                               ;SO LOCK OUT EXTRAINEOUS INTERRUPTS
24985  131674 012700 131572                 MOV     #DADTBL,R0      ;GET LOCATION OF FIRST REGISTER ADDRESS
24986  131700 012703 131572         1$:     MOV     #DADTBL,R3      ;MAKE R3 POINT TO LOCATION OF FIRST
24987                                                               ;REGISTER ADDRESS
24988  131704 012701 000005                 MOV     #5,R1           ;SET LOOP COUNTER TO CLEAR ALL REG.
24989  131710 005033                2$:     CLR     @(R3)+          ;CLEAR A REGISTER
24990  131712 077102                        SOB     R1,2$           ;LOOP UNTIL ALL REGISTERS CLEARED
24991  131714 012770 000100 000000          MOV     #BIT6,@(R0)     ;SET TEST BIT IN DEVICE REGISTERS
24992  131722 012701 131572                 MOV     #DADTBL,R1      ;GET LOCATION OF FIRST REGISTER ADDRESS
24993  131726 012702 000005                 MOV     #5,R2           ;SET UP TEST LOOP COUNTER
24994  131732 032731 000100         3$:     BIT     #BIT6,@(R1)+    ;IS TEST BIT SET IN THIS REGISTER
24995  131736 001006                        BNE     5$              ;IF YES GO SEE IF THERE IS AN ERROR
24996  131740 077204                4$:     SOB     R2,3$           ;LOOP UNTIL ALL REGISTER CHECKED
24997  131742 005030                        CLR     @(R0)+          ;CLEAR REGISTER JUST TESTED AND POINT
24998                                                               ;TO NEXT ONE
24999  131744 020027 131602                 CMP     R0,#TBLEND              ;ARE WE DONE TESTING
25000  131750 001407                        BEQ     7$              ;IF YES GO TO NEXT TEST
25001  131752 000752                        BR      1$              ;CONTINUE TESTING
25002  131754 021041                5$:     CMP     (R0),-(R1)      ;DID WE COMPARE THE REGISTER TO ITSELF?
25003  131756 001401                        BEQ     6$
  (2)  131760 104000                        EMT                     ;WRITE TO 1 INTERNAL ADDRESS MODIFIED
```

L3



```
25004                                                           ;ANOTHER SO ADDRESS NOT UNIQUE
25005  131762  062701  000002          6$:     ADD     #2,R1    ;RESTORE POINTER
25006  131766  000764                  BR      4$               ;GET BACK IN TEST LOOP
25007  131770  005000                  7$:     CLR     R0       ;INITIALIZE LOOP COUNTER
25008  131772  005200                  8$:     INC     R0       ;INCREMEMT COUNTER
25009  131774  001376                  BNE     8$               ;  UNTIL LOOP COUNTER = 0
25010
25011
25012
25013                                   ;*************************************************************************************
  (2)                                   ;TEST 662       TEST ALL INTERNAL OPTIONS SIMULTANEOUSLY
  (3)                                   ;*************************************************************************************
  (2)  131776                           TS662:
25014  131776  032737  000001  001020  10$:    BIT     #1, @#$ENV       ;ARE WE RUNNING UNDER APT
25015  132004  001405                          BEQ     70$              ;IF NO DO TEST
25016  132006  005737  001006                  TST     @#$PASS          ;IS THIS FIRST PASS
25017  132012  001402                          BEQ     70$              ;IF YES DO TEST
25018  132014  000167  001140                  JMP     ENDPAS           ;IF NO THEN SKIP THIS TEST
25019  132020  000005                  70$:    RESET                    ;CLEAR EVERY BODY
25020  132022  012767  000340  045746          MOV     #340,PS          ;SET PROCESSOR PRIORITY TO 7
25021  132030  017767  174700  105534          MOV     @TVECT2,$TMP0
25022  132036  017767  174666  105530          MOV     @RVECT2,$TMP1
25023  132044  017767  172554  105524          MOV     @TVECT,$TMP2
25024  132052  017767  172542  105520          MOV     @RVECT,$TMP3
25025  132060  017767  172546  105514          MOV     @RTCVT,$TMP4
25026  132066  005067  000360                  CLR     XMTCT2
25027  132072  005067  000356                  CLR     RECCT2
25028  132076  005067  000354                  CLR     TICKS
25029  132102  012777  132220  174624          MOV     #XMIT2, @TVECT2  ;SET UP SLU2 TRANSMIT VECTOR
25030  132110  012777  000340  174620          MOV     #340, @TPSW2     ;AND PSW
25031  132116  012777  132254  174604          MOV     #REC2, @RVECT2   ;SET UP SLU2 RECEIVER VECTOR
25032  132124  012777  000340  174600          MOV     #340, @RPSW2     ;AND PSW
25033  132132  012777  132210  172472          MOV     #TICKER, @RTCVT  ;SET UP RTC VECTOR
25034  132140  012777  000340  172466          MOV     #340, @RTCPSW    ;AND PSW
25035  132146  052777  000100  174550          BIS     #BIT6, @TCSR2    ;ENABLE SLU2 XMIT INTERRUPT
25036  132154  052777  000100  174536          BIS     #BIT6, @RCSR2    ;ENABLE SLU2 RECEIVER INTERRUPT
25037  132162  012703  132560                  MOV     #BUF2, R3        ;SET UP RECEIVER BUFFER
25038  132166  052777  000100  172434          BIS     #BIT6, @LKS      ;ENABLE RTC INTERRUPTS
25039  132174  012701  177777          3$:     MOV     #-1, R1          ;INITIALIZE DATA FOR SLU2
25040  132200  005067  045572                  CLR     PS               ;DROP PROCESSOR PRIORITY TO 0
25041  132204  000001                  WAITIO: WAIT                     ;WAIT FOR INTERRUPT
25042  132206  000776                  BR      WAITIO
25043
25044  132210  005267  000242          TICKER: INC     TICKS            ;UPDATE COUNT
25045  132214  000167  000056                  JMP     IOHAND           ;GO TO INTERRUPT HANDLER
25046
25047  132220  005267  000226          XMIT2:  INC     XMTCT2           ;UPDATE XMIT INTERRUPT COUNT
25048  132224  005201                          INC     R1               ;UPDATE XMIT DATA
25049  132226  010177  174474                  MOV     R1, @TBUF2       ;SEND NEXT CHARACTER
25050  132232  026727  000214  000400          CMP     XMTCT2, #400     ;IF 256 CHARACTERS HAVE NOT
25051  132240  002403                          BLT     1$               ;BEEN TRANSFERRED CONTINUE
25052  132242  042777  000100  174454          BIC     #BIT6, @TCSR2    ;ELSE NO MORE XMIT INTERRUPTS
25053  132250  000167  000022          1$:     JMP     IOHAND           ;GO TO INTERRUPT HANDLER
25054
25055  132254  005267  000174          REC2:   INC     RECCT2           ;UPDATE RECEIVER INTERRUPT COUNT
25056  132260  005777  174436                  TST     @RBUF2           ;BIT 15 SETS IF ANY ERRORS OCCURRED
```

```
25057  132264  100002                        BPL     3$              ;IF BIT IS CLEAR NO ERRORS
25058  132266  000005                        RESET                   ;CLEAR THE WORLD - STOP ALL
25059                                                                 ;INTERRUPTS
25060  132270  104000                        EMT                     ;RECEIVER STATUS ERROR
25061  132272  117723  174424        3$:     MOVB    @RBUF2, (R3)+   ;GET DATA AND STORE IT
25062
25063  132276  026727  000154  000074  IOHAND: CMP     TICKS, #74      ;HAS 1 SEC ELAPSED
25064  132304  001401                        BEQ     1$              ;IF YES STOP TEST
25065  132306  000002                        RTI                     ;RETURN FROM INTERRUPT TO AWAIT NEXT
25066  132310  042777  000100  172276  1$:     BIC     #BIT6, @TCSR    ;IF YES STOP TRANSMISSIONS
25067  132316  042777  000100  174400        BIC     #BIT6, @TCSR2   ;
25068  132324  042777  000100  172276        BIC     #BIT6,  @LKS    ;TURN OFF LINE CLOCK
25069
25070  132332  106427  000000        WAITER: MTPS    #0              ;LOWER PRIORITY TO ALLOW TIME FOR RECEIVER TO FINISH
25071  132336  012705  140000                MOV     #-40000,R5      ;SET UP LOOP COUNTER
25072  132342  062705  000001        1$:     ADD     #1, R5          ;DO LOOP UNTIL R5 = 0
25073  132346  001375                        BNE     1$
25074  132350  000005                        RESET                   ;STOP EVERYONE SHOULD BE DONE
25075
25076
25077  132352  026767  000074  000074  CHECK2: CMP     XMTCT2, RECCT2  ;#OF XMIT INTERRUPTS = REC INTERRUPTS
25078  132360  001401                        BEQ     1$
  (2)  132362  104000                        EMT                     ;INTERRUPT COMPARISON ERROR
25079  132364  012703  132560        1$:     MOV     #BUF2, R3       ;INITIALIZE TO FIRST RECEIVED DATA
25080  132370  005001                        CLR     R1              ;INITIALIZE TO FIRST XMIT DATA
25081  132372  016704  000054                MOV     XMTCT2, R4      ;GET # OF BYTES TRANSFERRED
25082  132376  122301                2$:     CMPB    (R3)+, R1       ;IS RECEIVED DATA = EXPECTED DATA
25083  132400  001401                        BEQ     3$
  (2)  132402  104000                        EMT                     ;SLU2 DATA COMPARISON ERROR
25084  132404  005201                3$:     INC     R1              ;UPDATE TO NEXT GOOD DATA
25085  132406  077405                        SOB     R4,2$           ;LOOP UNTIL ALL DATA CHECKED
25086  132410  010777  105156  174316  FINIE:  MOV     $TMP0, @TVECT2  ;RESTORE VECTORS
25087  132416  016777  105152  174304        MOV     $TMP1, @RVECT2
25088  132424  016777  105146  172172        MOV     $TMP2, @TVECT
25089  132432  016777  105142  172160        MOV     $TMP3, @RVECT
25090  132440  016777  105136  172164        MOV     $TMP4, @RTCVT
25091  132446  000167  000506                JMP     ENDPAS          ;FINISHED TESTING GO TO END OF PASS
25092
25093  132452  000000                XMTCT2: .WORD   0
25094  132454  000000                RECCT2: .WORD   0
25095  132456  000000                TICKS:  .WORD   0
25096
25097  132460  004767  000744        ERROR8: JSR     PC,ABORT                ;ARE WE UNDER UFD ?
25098  132464  012737  000007  001002        MOV     #7,@#$FATAL             ;SET UP FATAL ERROR NUMBER
25099  132472  012767  000001  046300        MOV     #1,$MSGTY               ;SET FATAL ERROR FLAG
25100  132500  032737  000001  001020        BIT     #1,@#$ENV               ;UNDER APT ?
25101  132506  001004                        BNE     COMMLT                  ;YES
25102  132510  012700  132522                MOV     #COMMSG,R0
25103  132514  004767  000644                JSR     PC,TYPE
25104  132520  000777        COMMLT: BR      .
25105
25106  132522  040506  046111  051125  COMMSG: .ASCIZ  /FAILURE DURING COMMON TEST/<12><15>
       132530  020105  052504  044522
       132536  043516  041440  046517
       132544  047515  020116  042524
       132552  052123  006412     000
```

```
25107           132560                          .EVEN
25108
25109  132560   000200                  BUF2:   .BLKW   200
25110
25111
25112
25113
25114  133160   005327                  ENDPAS: DEC     (PC)+           ;DECREMENT TEST LOOP COUNTER
25115  133162   000001                  $EOPCT: .WORD   1
25116  133164   003051                          BGT     $DOAGN          ;IF COUNTER NOT 0 DO TEST AGAIN
25117  133166   005267  045614                  INC     $PASS           ;INCREMENT PASS COUNTER
25118  133172   042767  100000  045606          BIC     #100000,$PASS   ;DON'T LET IT BE NEGATIVE
25119  133200   016767  045620  177754          MOV     $USWR,$EOPCT    ;RESET TEST LOOP COUNTER
25120  133206   012700  133607                  MOV     #ENDMSG,R0      ;LET R0 POINT TO ENDPASS MESSAGE
25121  133212   004767  000146                  JSR     PC,TYPE         ;GO TYPE END PASS MESSAGE
25122  133216   016700  044620                  MOV     42,R0           ;GET MONITOR ADDRESS
25123  133222   001413                          BEQ     DOAGIN          ;IF = 0 NO MONITOR SO DON'T STOP
25124  133224   000005                          RESET                   ;IF MONITOR CLEAR THE WORLD
25125  133226   013737  001270  000030  $ENDAD: MOV     @#SAV30,@#30    ;GET BACK THE VECTOR
25126  133234   013737  001272  000032          MOV     @#SAV32,@#32    ;
25127  133242   004710                          JSR     PC,(R0)         ;GO TO MONITOR
25128  133244   000240                          NOP                     ;THESE THREE LOCATIONS RESERVED
25129  133246   000240                          NOP
25130  133250   000240                          NOP
25131  133252   013737  000004  037572  DOAGIN: MOV     @#4,@#$TMP0
25132  133260   012737  133276  000004          MOV     #1$,@#4
25133  133266   012737  000001  164000          MOV     #1,@#164000
25134  133274   000402                          BR      2$
25135  133276   062706  000004          1$:     ADD     #4,SP
25136  133302   013737  037572  000004  2$:     MOV     @#$TMP0,@#4
25137  133310   000137                  $DOAGN: JMP     @(PC)+          ;RETURN TO TEST AT LOCATION RESTRT
25138  133312   001404                          .WORD   RESTRT
25139
25140
25141
25142                                   ;*************************************************************************************
25143                                   ;*COMMON SUBROUTINES THAT ARE NEEDED BY THE PROGRAM
25144                                   ;*
25145                                   ;*************************************************************************************
25146
25147  133314   012737  133330  000024  PWRDN:  MOV     #PWRUP,@#24     ;SET UP POWER FAIL VECTOR FOR POWER UP
25148  133322   004567  000102                  JSR     R5,ABORT        ;GO SEE IF WE ARE UNDER UFD
25149  133326   000000                          HALT
25150
25151  133330   012737  133314  000024  PWRUP:  MOV     #PWRDN,@#24     ;SET UP POWER FAIL VECTOR FOR POWER DOWN
25152  133336   012706  001000                  MOV     #STBOT,SP       ;SET UP STACK
25153  133342   005737  000172                  TST     @#MTFLAG        ;ARE WE ON MULTI-OPTION TESTER
25154  133346   001004                          BNE     1$              ;IF YES SKIP TYPE OUT
25155  133350   012700  133570                  MOV     #PWRMSG,R0      ;POINT R0 TO POWER FAIL MESSAGE
25156  133354   004767  000004                  JSR     PC,TYPE         ;GO TYPE IT
25157  133360   000167  046020          1$:     JMP     RESTRT          ;GO RESTART TEST
25158
25159
25160  133364   132767  000040  045427  TYPE:   BITB    #40,$ENVM       ;TYPE OUTS DISABLED
25161  133372   001015                          BNE     3$              ;IF YES GET TO EXIT
25162  133374   132737  000100  000052          BITB    #100,@#52       ;UNDER UFD ?
```

```
133402  001011                          BNE     3$                  ;IF YES GET TO EXIT
133404  105737  177564          1$:     TSTB    @#TTCSR             ;TEST FOR PRINTER READY BIT
133410  100375                          BPL     1$                  ;IF NOT READY WAIT FOR IT
133412  112037  177566                  MOVB    (R0)+,@#TPB         ;WHEN READY PRINT A CHARACTER
133416  001372                          BNE     1$                  ;IF LAST CHARACTER NOT NULL CONTINUE TYPING
133420  105737  177564          2$:     TSTB    @#TTCSR             ;WAIT FOR PRINTER TO FINISH
133424  100375                          BPL     2$
133426  000207                  3$:     RTS     PC
                                ;UFD ABORT MACRO

                                .SBTTL  ABORT ROUTINE FOR LCP/ORION UFD MODE


133430  005767  045640          ABORT:  TST     UFDFLG              ;TEST FOR USER FRIENDLY MODE
133434  001454                          BEQ     NOABRT              ;IF NOT UFD THEN CONTINUE NORMAL OPERATION
133436  020027  000032                  CMP     R0,#32              ;IS IT A ↑Z ?
133442  001443                          BEQ     ABORTZ              ;JUST GO BACK TO CHAIN IF IT IS (NO ERROR)
133444  020027  000003                  CMP     R0,#3               ;IS IS A ↑C ?
133450  001404                          BEQ     ABORTC              ;BR TO LOAD ↑C ON XXDP+ STACK (NO ERROR)
133452  005767  045620                  TST     UQUIET              ;TEST FOR USER-QUIET MODE
133456  001443                          BEQ     NOABRT              ;IF FIELD-SERVICE MODE, CONTINUE NORMAL OPERATION
                                                                    ; BECAUSE FIELD-SERVICE MODE DOES NOT QUIT ON ERROR
133460  000422                          BR      ABORTE              ;SET DRSERR THEN LEAVE


133462  016767  045602  044340  ABORTC: MOV     SAV30,30            ;RESTORE EMT LOCATION (30)
133470  016767  045576  044334          MOV     SAV32,32            ;RESTORE EMT PRIORITY LOCATION (32)
133476  104043                          EMT     +43                 ;GET XXDP STACK LOC. INTO R0 FROM MONITOR
133500  105720                  1$:     TSTB    (R0)+               ;FIND END OF STACK
133502  001376                          BNE     1$
133504  112760  000057  177777          MOVB    #'/,-1(R0)          ;LOAD SLASH OVER ZERO
133512  112720  000136                  MOVB    #'↑,(R0)+           ;LOAD UPARROW
133516  112720  000103                  MOVB    #'C,(R0)+           ;LOAD C
133522  105010                          CLRB    (R0)                ;MAKE NEW END TO STACK
133524  000412                          BR      ABORTZ              ;NOW LEAVE
133526  016767  045536  044274  ABORTE: MOV     SAV30,30            ;RESTORE EMT LOCATION (30)
133534  016767  045532  044270          MOV     SAV32,32            ;RESTORE EMT PRIORITY LOCATION (32)
133542  104042                          EMT     +42                 ;GET DCA LOCATION INTO R0 FROM MONITOR
133544  012760  177777  000042          MOV     #-1,42(R0)          ;SET A -1 INTO LOCATION DRSERR IN MONITOR
133552  013700  000042          ABORTZ: MOV     @#42,R0             ;AND PUT THE MONITOR RETURN ADDRESS IN R0
133556  005037  000042                  CLR     @#42                ;CLEAR MONITOR RETURN FLAG
133562  000167  177440                  JMP     $ENDAD              ;RETURN TO MONITOR-DO NOT PUSH STACK HERE
133566  000207                  NOABRT: RTS     PC                  ;IF NOTUFD RETURN TO MAINLINE


                                ;*******************************************************************************
                                ;*MESSAGES
                                ;*
                                ;*******************************************************************************

133570  047520  042527  020122  PWRMSG: .ASCIZ  /POWER FAILED/<12><15>
133576  040506  046111  042105
133604  006412     000
133607     105  042116  047440  ENDMSG: .ASCIZ  /END OF PASS CJKL5B0/<12><15>
133614  020106  040520  051523
```

```
                133622  041440  045512  032514
                133630  030102  006412     000
25180           133635     012  051415  040524  STRMSG: .ASCIZ  <12><15>/START TESTING/<12><15>
                133642  052122  052040  051505
                133650  044524  043516  006412
                133656     000
25181
25182
25183                   000001                  .END
```

```
A       020640          10298#  10300*  10301   10304   10307   10313*  10314   10317   10320   10326*  10327*  10328*  10329
                        10332   10335   10338   10342   10346*  10347*  10348   10349*  10352   10355   10360   10363   10381
                        10383   10387*  10388   10390   10394*  10395*  10396   10400*  10401*  10402
AAADON  074206          18345   18405#
AAA1    073406          18186#
AAA10   073736          18290#
AAA11   073766          18304#
AAA12   074016          18318#
AAA13   074046          18332#
AAA2    073436          18194#
AAA3    073466          18204#
AAA4    073516          18214#
AAA5    073546          18225#
AAA6    073576          18235#
AAA7    073626          18249#
AAA8    073656          18263#
AAA9    073706          18277#
AABBF0  110204          21330   21334   21341   21347   21355#
AABDON  110224          21349   21364#
AABTP1  110214          21329   21360#
AAB2    110202          21343   21346   21348   21352#
AACDON  115046          22514   22522#
AACTP1  115040          22502   22510   22518#
AAC10   115044          22505   22511   22513   22520#
AAC2    115014          22504   22506#
AADAT0  065670          16872   16890   16911   16919#
AADBF   124164          24062*  24063   24065   24068#
AADDON  124170          24066   24071#
AADONE  065770          16918   16951#
AAD1    124110          24056#
AAPAT0  065700          16866   16885   16902   16923#
AAPAT1  065710          16869   16887   16927#
AAPAT2  065720          16874   16931#
AAPAT3  065730          16892   16935#
AAPAT4  065740          16939#
AAPAT5  065750          16905   16943#
AAPAT6  065760          16913   16947#
AA11    065560          16888#
AA12    065562          16890#
AA13    065600          16894#  16896
AA2     065510          16870#
AA20    065606          16895   16896#
AA22    065632          16906#
AA23    065634          16908#
AA24    065654          16915#  16917
AA27    065662          16916   16917#
AA3     065512          16872#
AA4     065530          16876#  16878
AA7     065536          16877   16878#
ABASE = 000000           6424
ABORT   133430           6633   11058   14340   14485   14778   14983   14995   24130   24367   24507   24952   25097   25148
                        25172#
ABORTC  133462          25172#
ABORTE  133526          25172#
ABORTZ  133552          25172#
ACDW1 = 000000           6424
```

```
ACD#2 = 000000          6424
ACPUOP= 000000          6424
ACTSUM  054130         14904#  14917*  14921*  14946   14956
ADC1    014604          9385    9386    9387#
ADC2    014606          9387    9388#
ADC3    014626          9392    9393    9394#
ADC4    014630          9394    9395#
ADC5    014646          9398    9399    9400    9401#
ADDW0 = 000000          6424
ADDW1 = 000000          6424
ADDW10= 000000          6424
ADDW11= 000000          6424
ADDW12= 000000          6424
ADDW13= 000000          6424
ADDW14= 000000          6424
ADDW15= 000000          6424
ADDW2 = 000000          6424
ADDW3 = 000000          6424
ADDW4 = 000000          6424
ADDW5 = 000000          6424
ADDW6 = 000000          6424
ADDW7 = 000000          6424
ADDW8 = 000000          6424
ADDW9 = 000000          6424
ADD1    014466          9353    9354    9355#
ADD2    014470          9355    9356#
ADD3    014504          9359    9360    9361#
ADD4    014506          9361    9362#
ADD5    014524          9365    9366    9367#
ADD6    014526          9367    9368#
ADD7    014540          9369    9370    9371#
ADD8    014542          9371    9372#
ADD9    014562          9375    9376    9377    9378#
ADEVCT= 000000          6424
ADEVM = 000000          6424
ADONE   054742         15131#
AENV  = 000000          6424
AENVM = 000000          6424
AERR1   054732         15110   15111   15112   15127#
AFATAL= 000000          6424
AMADR1= 000000          6424
AMADR2= 000000          6424
AMADR3= 000000          6424
AMADR4= 000000          6424
AMAMS1= 000000          6424
AMAMS2= 000000          6424
AMAMS3= 000000          6424
AMAMS4= 000000          6424
AMSGAD= 000000          6424
AMSGLG= 000000          6424
AMSGTY= 000000          6424
AMTYP1= 000000          6424
AMTYP2= 000000          6424
AMTYP3= 000000          6424
AMTYP4= 000000          6424
ANSR    054122         14901#
```

```
APASS = 000000           6424
APRIOR= 000000           6424
APTENV= 000001           6404#  10956
AROUN   001740           6564    6567#
AROUND  025256          11071   11082#
ASLB1   016164           9735    9736#
ASLB2   016152           9731    9732    9733#
ASLB3   016154           9733    9734#
ASLB4   016176           9737    9738    9739    9740#
ASL1    015556           9624    9625    9626    9627#
ASL2    015560           9627    9628#
ASL3    015576           9631    9632    9633    9634#
ASL4    015600           9634    9635#
ASL5    015614           9638    9639    9640#
ASL6    015616           9640    9641#
ASL7    015642           9644    9645    9646    9647    9649#
ASRB1   016222           9749    9750#
ASRB2   016234           9751    9752    9753    9754#
ASRB3   016236           9754    9755#
ASRB4   016244           9756    9757#
ASRB5   016246           9757    9758#
ASRB6   016266           9761    9762    9763    9764#
ASRB7   016270           9764    9765#
ASR1    015666           9655    9656    9657    9658#
ASR2    015670           9658    9659#
ASR3    015712           9663    9664    9665    9666#
ASR4    015714           9666    9667#
ASR5    015730           9670    9671    9672    9673#
ASR6    015732           9673    9674#
ASR7    015762           9678    9679    9680    9681    9683#
ASTART  025746          11351#  11543   11547
ASWREG= 000000           6424
ATESTN= 000000           6424
ATRAP   023652          10825   10841#
AUNIT = 000000           6424
AUSWR = 000000           6424
AUT01   023744          10864#
AVECT1= 000000           6424
AVECT2= 000000           6424
A1      054702          15116#  15130
A11     054702          15117#
A12     054716          15122#
A2      054734          15127   15129#
BA      051760          14513#  14550*  14551*  14676*
BBBDON  074462          18486   18493#
BBBER1  074256          18423   18435#
BBBP1   074442          18424   18426   18446   18467   18488#
BBBP2   074452          18444   18463   18489#
BBB0    074212          18419#
BBB1    074240          18428#
BBB2    074272          18438   18441#
BBB3    074312          18447#
BBB4    074344          18457   18460#
BBB5    074372          18469#  18472
BBB6    074400          18466   18472#
BBB7    074260          18435   18436#
```

```
BBCDON  115122          22540   22551#
BBCTP1  115114          22528   22536   22545#
BBC10   115120          22530   22537   22539   22550#
BBC2    115070          22532#
BBDAT0  067250          17137   17155   17174   17192   17209   17226   17235#
BBDDON  124420          24114   24123#
BBDONE  067400          17234   17279#
BBD1    124200          24077#
BBPAT0  067260          17165   17176   17239#
BBPAT1  067270          17134   17152   17170   17188   17206   17223   17243#
BBPAT2  067300          17132   17139   17247#
BBPAT3  067310          17184   17251#
BBPAT4  067320          17150   17255#
BBPAT5  067330          17204   17259#
BBPAT6  067340          17221   17228   17263#
BBP10   067360          17157   17271#
BBP11   067370          17211   17275#
BBP7    067350          17194   17267#
BB10    066726          17159#  17161
BB13    066734          17160   17161#
BB14    066744          17163   17165#
BB15    066764          17171#
BB16    067010          17178#  17180
BB17    067016          17179   17180#
BB2     066624          17135#
BB20    067026          17182   17184#
BB21    067046          17189#
BB22    067072          17196#  17198
BB25    067100          17197   17198#
BB26    067110          17200   17202#
BB27    067130          17207#
BB3     066630          17137#
BB30    067152          17213#  17215
BB31    067160          17214   17215#
BB32    067170          17217   17219#
BB33    067210          17224#
BB34    067232          17230#  17232
BB35    067240          17231   17232#
BB4     066646          17141#  17143
BB5     066654          17142   17143#
BB6     066664          17146   17148#
BB7     066704          17153#
BCF     054116          14899#  14916   14919   14940*  14950*
BDONE   054776          15153#
BDVHLT  054442           6527*  14987   14990#  14999
BDVMSG  054444          14988   14992#
BDVTST  053514          14535   14704   14773   14804#
BDVTS2  054176          14878   14935#
BELL  = 000240           6387#
BIC1    014046           9199    9200    9201#
BIC2    014050           9201    9202#
BIC3    014066           9205    9206    9207#
BIS1    014112           9213    9214    9215    9216#
BIS2    014114           9216    9217#
BIS3    014134           9220    9221    9222    9223#
BITCHK  020314         10196#
```

```
BITCLR  020242          10182#
BITCON  020370          10209   10210#
BITSET  020260          10189#
BITST1  014002           9184    9185    9186#
BITST2  014004           9186    9188#
BITST3  014022           9191    9192    9193#
BIT0  = 000001          12087#  12847   12848   12849   12858   12879   12909   12910   12911   12912   12958   14385   14413
                        14750   24157#  24606   24608   24609   24611   24612   24615   24617   24905   24919   24923
BIT1  = 000002          24158#
BIT10 = 002000          14809   24167#
BIT11 = 004000          24168#
BIT12 = 010000          24169#
BIT13 = 020000          24170#  24924
BIT14 = 040000          24171#  24889
BIT15 = 100000          14470   14858   14868   24172#  24891   24926
BIT2  = 000004          24159#
BIT3  = 000010          24160#
BIT4  = 000020          12849   12878   12909   12910   12911   12912   12957   14386   14746   24161#
BIT5  = 000040           6484   10953   24162#
BIT6  = 000100           6484   14533   24091   24105   24163#  24269   24271   24272   24274   24275   24278   24280   24287
                        24295   24300   24306   24312   24315   24324   24332   24341   24347   24351   24358   24399   24401
                        24402   24404   24405   24412   24414   24426   24445   24451   24468   24481   24493   24500   24625
                        24627   24628   24630   24631   24638   24640   24649   24651   24652   24654   24655   24662   24664
                        24671   24679   24684   24690   24696   24699   24708   24716   24725   24732   24736   24743   24785
                        24786   24795   24798   24811   24814   24832   24839   24851   24858   24869   24874   24877   24991
                        24994   25035   25036   25038   25052   25066   25067   25068
BIT7  = 000200          14680   24164#
BIT8  = 000400          14702   24165#
BIT9  = 001000          14443   24166#
BRCT    001764           6554*   6578#   6581*
BRH     001750           6555*   6572#
BRTAB   021432           6552   10449#
BR1     021634          10527   10529#
BR10    021776          10566   10568#
BR11    022056          10579   10581#
BR12    022116          10587   10589#
BR13    022156          10595   10597#
BR14    022216          10603   10605#
BR15    022270          10614   10616#
BR16    022302          10617   10619#
BR17    022314          10620   10622#
BR2     021654          10532   10534#
BR20    022326          10623   10624#
BR21    022340          10625   10627#
BR22    022352          10628   10630#
BR23    022364          10631   10633#
BR3     021670          10537   10539#
BR4     021706          10543   10545#
BR5     021724          10549   10551#
BR6     021742          10555   10556#
BR7     021760          10560   10562#
BR70    023472          10796#
BR71    024206          10899   10905#
BTCON   020426          10223   10226#
BTERR   020420          10224#
BTRAP   023720          10849   10860#
```

```
BUF2    132560          25037   25079   25109#
B1      054752          15142#  15152
B2      054754          15145#
B3      054774          15151   15152#
CARRY1  003456           6733    6734#
CC    = 177776          6399#  11106#
CCBDON  110264          21382   21385#
CCB10   110262          21379   21381   21384#
CCB2    110236          21374#
CCCDON  075214          18612   18668#
CCC:    074466          18507#
CCC10   074752          18580#
CCC11   074776          18588#
CCC12   075022          18597#
CCC13   075046          18605#
CCC2    074512          18515#
CCC3    074536          18523#
CCC4    074562          18530#
CCC5    074606          18538#
CCC6    074632          18546#
CCC7    074656          18555#
CCC8    074702          18564#
CCC9    074726          18572#
CCDAT0  066440          16971   16988   17006   17024   17041   17058   17067#
CCERR   020612          10282#
CCP0    066450          16966   16983   16998   17036   17053   17071#
CCP1    066460          16985   17075#
CCP10   066550          17026   17103#
CCP11   066560          17043   17107#
CCP12   066570          17111#
CCP2    066470          16968   16973   17079#
CCP3    066500          17016   17038   17083#
CCP4    066510          17055   17060   17087#
CCP5    066520          17020   17091#
CCP6    066530          17002   17008   17095#
CCP7    066540          16990   17099#
CCXDON  066600          17066   17116#
CCX12   066124          16993   16994#
CCX13   066134          16996   16998#
CCX14   066154          17003#
CCX15   066200          17010#  17012
CCX18   066206          17011   17012#
CCX19   066216          17014   17016#
CCX2    066014          16969#
CCX20   066236          17021#
CCX21   066262          17028#  17030
CCX24   066270          17029   17030#
CCX25   066300          17032   17034#
CCX26   066320          17039#
CCX27   066342          17045#  17047
CCX3    066036          16975#  16977
CCX30   066350          17046   17047#
CCX31   066360          17049   17051#
CCX32   066400          17056#
CCX33   066422          17062#  17064
CCX36   066430          17063   17064#
```

```
CCX6    066044          16976   16977#
CCX7    066054          16979   16981#
CCX8    066074          16986#
CCX9    066116          16992#  16994
CC1     001744           6557*   6559*   6570#
CC2     020522          10258*  10266#  10268*  10274*
CC3     020506          10257*  10262#  10269*  10270   10272
CDONE   055226          15233   15234#
CHECK2  132352          25077#
CHKAPT  050622          12525   14327#  24077
CHKSUM  054132          14915#  14944   14954
CIS     025544          11094   11146#
CISADR  025250          11077#  11086   11087
CISTRP  025234          11074#  11084
CLM     001316           6486    6489#
CLRCD   020504          10261#  10271   10275
CLR1    014342           9303    9304    9305    9306#
CMPREG  050440          12709   12716   12723   12730   14267#
CMPSUB  074102          18186   18194   18204   18214   18225   18235   18249   18263   18277   18290   18304   18318   18332
                        18371#
CMPTMP  074176          18391   18401#
CMP1    014764           9444    9445    9446#
CMP2    014766           9446    9447#
CMP3    015010           9451    9452    9453#
CMP4    015012           9453    9454#
CMP5    015036           9458    9459    9460    9461#
CMP6    015040           9461    9462#
CMP7    015060           9465    9466    9467#
CNTR    050752          14363#  14388*  14389   14397*  14448*  14449*  14479   14480   14676   14677*  14678*  14683   14685
                        14710*  14722   14748*  14749*  14751   14759   14764*  14766*
COMHLT  132520           6526*  25101   25104#
COMMSG  132522          25102   25106#
COMPLE  054374          14969#
COM1    015102           9474    9475    9476#
CONCIS  025310          11075   11090#
CONT    001766           6562    6566    6579#
CONTIN  001376           6495    6497    6500#
CONT42  131324          24908   24912#
CON1    020530          10267   10268#
CON2    020614          10282   10283#
CORH    023712          10842*  10855#
COUNT   025624          11177#  11178   11324*  11368*  11371   11400*  11403   11431*  11434   11462*  11465   11494*  11498
                        11527*  11530   11639*  11666*  11674   11705*  11712   11744*  11751
COUNTR  054120          14900#  14938*  14962*
CPUHLT  002346           6518*   6628    6637    6640#
CPUMSG  002350           6638    6642#
CSRMSG  051673          14492   14501#
CSR1    051754          14511#  14547*  14548   14550   14552   14554   14556   14558   14560   14587*  14609*  14627*  14646*
                        14673*
CSR2    051756          14512#  14548*  14549*  14589*  14591*  14597*  14599*  14610*  14612*  14617*  14619*  14628*  14630*
                        14635*  14637*  14647*  14649*  14654*  14656*  14658*  14659*  14674*  14680   14714*  14717*
CTRAP   023670          10847#  10864
C15     055006          15169#
C2      055022          15174   15175#
C25     055030          15178#
C3      055046          15183   15184#
```

```
C35     055054          15187#
C4      055072          15192   15193#
C45     055076          15195#
C5      055114          15200   15201#
C55     055120          15204#
C6      055134          15209   15210#
C65     055142          15212#
C7      055160          15217   15218#
C75     055166          15220#
C8      055204          15225   15226#
C85     055210          15228#
DADTBL  131572          24965#  24985   24986   24992
DAERR   020366          10197   10199   10201   10203   10205   10207   10209#
DALTB1  044340          13235   13288#
DALTB2  044374          13249   13306#
DALTB3  044624          13334   13387#
DALTB4  044660          13348   13405#
DATA    051764          14515#  14554*  14555*  14672*
DBE     025052          11032#  11035
DBE1    025060          11027   11035#
DBE2    025056          11029   11034#
DBE3    025072          11036   11037#
DBE4    025104          11038   11039#
DBE5    025114          11040   11041#
DDBBFO  110450          21438   21460#
DDBDON  110460          21447   21465#
DDBTP1  110430          21424   21451#
DDBTP2  110440          21428   21455#
DDB2    110364          21433#
DDB5    110426          21443   21446   21448#
DDCDON  115222          22567   22573#
DDCTP1  115206          22557   22563   22570#
DDC10   115220          22560   22564   22566   22572#
DDC2    115154          22559   22561#
DDDAT0  070150          17296   17314   17332   17350   17367   17384   17402   17412#
DDDDON  075706          18741   18801#
DDDONE  070300          17411   17456#
DDD1    075220          18681#
DDD2    075260          18689#
DDD3    075320          18697#
DDD4    075360          18705#
DDD5    075420          18713#
DDD6    075460          18723#
DDD7    075520          18732#
DDONE   055374          15279   15300#
DDP0    070160          17316   17334   17416#
DDP1    070170          17291   17293   17311   17327   17420#
DDP2    070200          17309   17329   17424#
DDP3    070210          17345   17364   17428#
DDP4    070220          17379   17399   17432#
DDP5    070230          17381   17397   17436#
DDP6    070240          17347   17362   17440#
DDP7    070250          17352   17369   17444#
DDP8    070260          17386   17404   17448#
DDP9    070270          17298   17452#
DD10    067532          17318#  17320
```

```
DD11    067540          17319   17320#
DD12    067554          17323   17325#
DD13    067574          17330#
DD14    067616          17336#  17338
DD15    067624          17337   17338#
DD16    067640          17341   17343#
DD17    067660          17348#
DD18    067702          17354#  17356
DD2     067424          17294#
DD21    067710          17355   17356#
DD22    067720          17358   17360#
DD23    067740          17365#
DD24    067762          17371#  17373
DD27    067770          17372   17373#
DD3     067446          17300#  17302
DD30    070000          17375   17377#
DD31    070020          17382#
DD32    070042          17388#  17390
DD35    070050          17389   17390#
DD36    070064          17393   17395#
DD37    070104          17400#
DD38    070126          17406#  17408
DD41    070134          17407   17408#
DD6     067454          17301   17302#
DD7     067470          17305   17307#
DD8     067510          17312#
DEC1    014242           9264    9265    9266    9267#
DEC2    014244           9267    9268#
DEC3    014260           9271    9272    9273#
DEC4    014262           9273    9274#
DEC5    014276           9277    9278    9279#
DEC6    014300           9279    9280#
DEC7    014322           9284    9285    9286    9287#
DERR1   055330          15259   15283#
DERR2   055276          15258   15271#
DEVADR  051752          14510#  14543*  14545   14547   14564*
DEVECT  051746          14508#  14544*  14565*  14566   14569   14586*  14595*  14607*  14615*  14626*  14633*  14645*  14652*
DEV1    051750          14509#  14543
DISPRE  000174           6455#
DIVDSU  075564          18681   18689   18697   18705   18713   18723   18732   18770#
DIVDT   075676          18786   18790   18799#
DIVFSU  075076          18507   18515   18523   18530   18538   18546   18555   18564   18572   18580   18588   18597   18605
                        18640#
DIVFT   075204          18656   18666#
DLOOP   054234          14940#  14963
DL11W   024012          10873   10876#
DN*B0A  007662           8154    8155    8156    8157#
DN*B0B  007664           8157    8158#
DN*B2A  010016           8193    8194    8195    8196#
DN*B2B  010020           8196    8197#
DN*B2C  010026           8198    8199#
DN*B2D  010042           8201    8202    8203#
DN*B2E  010044           8203    8204#
DN*B2F  010054           8206    8207#
DN*B3A  010120           8220    8221    8222    8223#
DN*B3B  010122           8223    8224#
```

```
DNMB3C  010132          8225    8226#
DNMB3D  010150          8229    8230    8231#
DNMB3E  010152          8231    8232#
DNMB4A  010272          8259    8260    8261    8262#
DNMB4B  010274          8262    8263#
DNMB4C  010304          8264    8265#
DNMB4D  010314          8266    8267#
DNMB4E  010316          8267    8268#
DNMB4F  010324          8269    8270#
DNM03A  007262          7956    7957    7958    7959#
DNM03B  007264          7959    7960#
DNM03C  007274          7961    7963#
DNM1    007174          7931    7932#
DNM1A   007714          8166    8167    8168    8169#
DNM1B   007716          8169    8170#
DNM2    007202          7933    7934#
DNM2A   007746          8178    8179    8180#
DNM2B   007750          8180    8181#
DNM2C   007756          8183#
DNM2D   007760          8183    8184#
DNM3    007212          7936    7937#
DNM4    007226          7941    7942#
DNM4A   010216          8243    8244    8245    8246#
DNM4B   010220          8246    8247#
DNM4C   010226          8248    8249#
DNM5A   010366          8280    8281    8282    8283#
DNM5B   010370          8283    8284#
DNM5C   010400          8285    8286#
DNM6A   010442          8295    8296    8297    8298#
DNM6B   010444          8298    8299#
DNM6C   010454          8300    8301#
DNM7A   010520          8310    8311    8312    8313#
DNM7B   010522          8313    8314#
DNM7C   010532          8315    8316#
DOAGIN  133252         25123   25131#
DOPB2A  007502          8086    8087#
DOPB2B  007534          8104    8105#
DOP0A   007030          7877    7878#
DOP0B   007046          7883    7884#
DOP0C   007060          7887    7888#
DOP0D   007102          7895    7896#
DOP03A  007142          7913    7914    7915    7916#
DOP03B  007144          7916    7917#
DOP1    007402          8031    8032#
DOP2    007452          8066    8067#
DOP4    011514          8550#
DOP5    011562          8574#
DPAT3   062122         16030#
DRLP    054376         14935*  14936   14974#
DTRAP1  023710         10853#  10863
DUMMY =  000000        11173#  11869   11870   11871   11872   11873   11874   11875
D1      055252         15261#  15275   15281
D2      055264         15264*  15265#
D3      055266         15266#  15283
D4      055272         15268#
D5      055300         15272#  15296
```

```
D6      055314          15273   15277#
D7      055324          15278   15280#
D8      055360          15290#
D9      055372          15295   15298#
EDONE   055442          15323   15324#
EEBBF0  110574          21498#
EEBBF1  110604          21470   21476   21481   21487   21502#
EEBDON  110616          21491   21507#
EEBTP1  110564          21469   21494#
EEB10   110614          21477   21484   21488   21490   21506#
EEB2    110524          21478#
EECDON  115330          22585   22595#
EECTP1  115304          22576   22577*  22590#  22592
EECTP2  115314          22575   22581   22592#
EEC10   115326          22578   22582   22584   22594#
EEC2    115252          22579#
EEDAT0  070454          17473   17491   17501#
EEDONE  070536          17500   17526#
EEEDON  076440          18918   18974#
EEE1    075712          18815#
EEE10   076176          18887#
EEE11   076222          18895#
EEE12   076246          18903#
EEE13   076272          18911#
EEE2    075736          18823#
EEE3    075762          18831#
EEE4    076006          18839#
EEE5    076032          18847#
EFE6    076056          18855#
EEE7    076102          18863#
EEE8    076126          18871#
EEE9    076152          18879#
EEP0    070464          17475   17493   17505#
EEP1    070476          17468   17470   17510#
EEP2    070506          17514#
EEP3    070516          17486   17488   17518#
EEP4    070526          17522#
EERR0   055440          15310   15319   15323#
EE12    070440          17496   17497#
EE2     070324          17471#
EE3     070346          17477#  17479
EE6     070354          17478   17479#
EE7     070370          17482   17484#
EE8     070410          17489#
EE9     070432          17495#  17497
EISEND  037552          12071   12072#
EMTA  = 104377           6402#   6646
EMTSAV  001202           6484#
ENDMSG  133607          25120   25179#
ENDPAS  133160          24974   24977   25018   25091   25114#
ENT176  031434          11856   11864#
ENT51   027360          11602   11606#
ER      001762           6563    6565    6577#
ERRORA  051550          14374   14485#
ERRORB  053402          14537   14778#
ERRORC  054402          14815   14983#
```

```
ERRORD  054502          14807  14995#
ERRORE  051742          14447* 14481* 14490  14504#
ERROR1  002306           6577   6626  6627   6633#  6645  6646  6652  6676  10678 10679 10686 10687 10689
                        10690  10691  10694
ERROR2  002650           6647   6652# 6655   6662   6663
ERROR3  050646          12286  14340#
ERROR4  124452          15094  24130# 24147
ERROR5  125700          24193  24367#
ERROR6  126624          24385  24507#
ERROR7  131474          24534  24952#
ERROR8  132460          24979  25097#
ERRVEC- 000004          12088# 13044* 13056* 14380  14381  14382* 14383* 14399* 14400* 14414  14415  14416* 14417*
                        14427* 14428* 14448  14449  14450* 14479* 14480* 14541  14542* 14562* 15112* 15258* 15352*
                        15381* 15840* 16120* 16176* 16236* 18045* 20674* 20731* 20788* 20837* 20880* 20922* 21477*
                        21519* 21558* 21600* 21641* 21683* 21720* 21759* 21794* 22505* 22530* 22560* 22578* 22600*
                        22622* 22645* 22674* 23380* 23409* 23438* 23466* 23497* 23527* 23558* 24148*
ESR0    051132          14382  14396#
ESR1    051074          14389# 14398
ESR10   053570          14810  14813  14815#
ESR11   052062          14545# 14567
ESR12   053116          14708  14717#
ESR13   053130          14712  14720#
ESR14   053132          14719  14722#
ESR15   053270          14742  14757#
ESR16   053346          14752  14769#
ESR17   053250          14751# 14767
ESR18   053324          14760  14764#
ESR19   051502          14473# 14475
ESR2    051144          14390  14399#
ESR20   053030          14703  14706#
ESR21   052336          14586  14595#
ESR22   052372          14595  14602#
ESR23   052374          14601  14607#
ESR24   052436          14607  14615#
ESR25   052770          14681  14683#
ESR26   052756          14680# 14682
ESR27   052472          14615  14622#
ESR28   052474          14621  14626#
ESR29   052532          14626  14633#
ESR3    051014          14377  14380#
ESR30   052570          14639  14645#
ESR31   052566          14633  14640#
ESR32   053010          14684  14687#
ESR33   052626          14645  14652#
ESR34   051616          14491  14493  14494#
ESR35   052702          14652  14662#
ESR36   126244          24429#
ESR37   126262          24430  24433#
ESR38   126300          24433  24436  24437#
ESR39   126274          24435#
ESR4    051300          14416  14425#
ESR40   053616          14819  14821#
ESR41   053640          14824  14826#
ESR42   053662          14829  14831#
ESR43   053702          14834  14836#
ESR44   053722          14839  14841#
```

```
ESR45   053744          14844   14846#
ESR46   053764          14850   14852#
ESR47   054004          14855   14857#
ESR48   054032          14861   14863#
ESR49   054044          14864   14867#
ESR5    051302          14422   14426#
ESR50   054074          14871   14873#
ESR51   126260          24424   24432#
ESR52   126312          24440#
ESR53   126324          24443#
ESR54   126346          24447#
ESR55   126352          24444   24450#
ESR57   051340          14444   14447#
ESR6    051414          14455   14458#
ESR7    051416          14453   14459#
ESR8    051472          14460   14470#
ESR9    051452          14465#  14471
ESR98   052256          14563   14569#
ESR99   052230          14542   14564#
EXADHT  051626           6520*  14489   14496#
EXATST  050756          12285   14189   14372#
EXPSUM  054126          14903#  14945*  14946*  14947   14955*  14956*  14957
EXTMSG  051630          14494   14499#
E1      055406          15312#
E3      055414          15314#
E4      055416          15315#
F     - 000063          11174#  11613#  11614#  11615#  11616#  11617#  11618#  11619#  11620#  11621#  11622#
FDATIO  055716          15344   15364   15404#
FDATI1  055720          15405#
FDATI2  055722          15406#
FDATI3  055724          15407#
FDATI4  055726          15351   15360   15408#
FDATI5  055730          15409#
FDATI6  055732          15410#
FDATI7  055734          15411#
FDATO0  055740          15376   15393   15413#
FDATO1  055742          15414#
FDATO2  055744          15415#
FDATO3  055746          15416#
FDATO4  055750          15380   15389   15417#
FDATO5  055752          15418#
FDATO6  055754          15419#
FDATO7  055756          15420#
FDONE   056002          15402   15432#
FERR20  055714          15352   15381   15402#
FFB8F1  110732          21512   21518   21524   21530   21541#
FFBDON  110744          21534   21546#
FFBTP1  110722          21511   21537#
FFB10   110742          21519   21527   21531   21533   21545#
FFB2    110662          21521#
FFCDON  115430          22607   22617#
FFCTP1  115406          22598   22599*  22612#  22614
FFCTP2  115416          22597   22603   22614#
FFC10   115426          22600   22604   22606   22616#
FFC2    115362          22601#
FFDATO  070702          17545   17564   17575#
```

```
FFDONE  070762          17572   17601#
FFFDON  076772          19020   19080#
FFF1    076444          18988#
FFF2    076504          18996#
FFF3    076544          19004#
FFF4    076604          19012#
FFP0    070712          17559   17580#
FFP1    070722          17561   17584#
FFP2    070732          17540   17588#
FFP3    070742          17542   17592#
FFP4    070752          17547   17566   17596#
FF10    070672          17569   17570#
FF2     070562          17543#
FF3     070604          17550#  17552
FF4     070612          17551   17552#
FF5     070622          17554   17557#
FF6     070642          17562#
FF7     070664          17568#  17570
FINIE   132410          25086#
FINISH  025610          11070*  11100   11164#
FOVER   023112          10698   10709#
FPEXIT  124446          24078   24127#
FPMLT   124512           6522*  24134   24137#
FPMSG   124514          24135   24139#
FPP     025600          11070   11098   11160#
FPSTRT  054616          14969   15089#
FPVECT= 000244          15077#  15110*  15259*  15310*  15835*  15863*  16407*  16480*  17795*  17815*  17846*  17866*  17896*
                        17916*  18423*  18466*  19181*  19314*  19443*  19583*  20498*  24147*
FXDAT0  055762          15345   15365   15394   15422#
FXDAT1  055764          15423#
FXDAT2  055766          15424#
FXDAT3  055770          15425#
FXDAT4  055772          15426#
FXDAT5  055774          15427#
FXDAT6  055776          15428#
FXDAT7  056000          15429#
F10     055630          15385#
F11     055632          15386#
F12     055646          15390   15391#
F13     055672          15396#  15398
F135    055656          15392   15393#
F14     055700          15397   15398#
F2      055470          15348#  15349
F22     055702          15399#
F3      055510          15356#
F4      055512          15357#
F5      055526          15361   15362#
F6      055574          15373   15374#
F7      055610          15377#  15378
GAND0   060370          15714#
GAND1   060372          15715#
GAND2   060374          15716#
GAND3   060376          15717#
GCMP    060320          15647   15648   15649   15650   15651   15652   15653   15654   15655   15656   15657   15658   15690#
GDAT00  060410          15647   15648   15649   15650   15651   15652   15653   15654   15655   15656   15657   15658   15677
                        15683   15690   15724#
```

```
GDATO1  060412          15725#
GDATO2  060414          15726#
GDATO3  060416          15727#
GDONE   060420          15660   15730#
GFLAG1  060344          15647*  15648*  15649*  15650*  15651*  15652*  15653*  15654*  15655*  15656*  15657*  15658*  15663
                        15701#
GFLAG2  060346          15702#
GGBBFO  111060          21551   21563   21569   21580#
GGBBF1  111070          21557   21584#
GGBDON  111102          21573   21589#
GGBTP1  111050          21550   21576#
GGB10   111100          21558   21566   21570   21572   21588#
GGB2    111010          21560#
GGCDON  115524          22631   22639#
GGCTP1  115510          22619   22620*  22627   22636#
GGC10   115522          22622   22626   22628   22630   22638#
GGC2    115456          22623#
GGDATO  072420          17801   17827   17852   17877   17902   17928   17943#
GGDONE  072540          17942   17985#
GGERO   071666          17795   17821#  17846
GGER14  072324          17896   17922#
GGGDON  101556          19757   19834#
GGG1    100626          19633#
GGG10   101156          19714#
GGG11   101206          19723#
GGG12   101236          19732#
GGG13   101266          19741#
GGG14   101316          19750#
GGG2    100656          19642#
GGG3    100706          19651#
GGG4    100736          19660#
GGG5    100766          19669#
GGG6    101016          19678#
GGG7    101046          19687#
GGG8    101076          19696#
GGG9    101126          19705#
GGP1    072430          17948#
GGP2    072440          17847   17867   17897   17917   17952#
GGP3    072450          17849   17869   17956#
GGP4    072460          17960#
GGP5    072470          17796   17798   17816   17818   17964#
GGP6    072500          17803   17829   17854   17904   17968#
GGP7    072510          17879   17973#
GGP8    072520          17899   17919   17977#
GGP9    072530          17930   17981#
GG10    071762          17841   17844#
GG11    072010          17850#
GG12    072032          17856#  17858
GG13    072040          17857   17858#
GG14    072054          17861   17864#
GG15    072102          17870#  17872
GG16    072106          17866   17872#
GG17    072142          17881#  17883
GG18    072150          17882   17883#
GG19    072200          17891   17894#
GG2     071570          17799#
```

```
GG20    072226          17900#
GG21    072250          17906#  17908
GG22    072256          17907   17908#
GG23    072272          17911   17914#
GG24    072320          17920#  17923
GG25    072326          17916   17923#
GG26    072362          17932#  17934
GG27    072370          17933   17934#
GG3     071612          17805#  17807
GG4     071620          17806   17807#
GG5     071634          17810   17813#
GG6     071662          17819#  17822
GG7     071670          17815   17822#
GG8     071724          17831#  17833
GG9     071732          17832   17833#
GIN1    025322          11092#  11099   11121
GIN2    025364          11103#  11122
GIN3    025412          11101   11108#
GOR0    060400          15719#
GOR1    060402          15720#
GOR2    060404          15721#
GOR3    060406          15722#
GPAT00  060350          15647   15649   15651   15653   15655   15657   15673   15704#
GPAT01  060352          15705#
GPAT02  060354          15706#
GPAT03  060356          15707#
GPAT10  060360          15648   15650   15652   15654   15656   15658   15668   15709#
GPAT11  060362          15710#
GPAT12  060364          15711#
GPAT13  060366          15712#
GRESET  060300          15647   15648   15649   15650   15651   15652   15653   15654   15655   15656   15657   15658   15683#
GSETUP  060222          15647   15648   15649   15650   15651   15652   15653   15654   15655   15656   15657   15658   15663#
GS1     060252          15673#  15687
G1      056636          15647#
G10     057070          15649#
G11     057072          15649#
G12     057136          15650#
G13     057170          15650#
G14     057172          15650#
G15     057236          15651#
G16     057270          15651#
G17     057272          15651#
G2      056670          15647#
G20     057336          15652#
G21     057370          15652#
G22     057372          15652#
G23     057436          15653#
G24     057470          15653#
G25     057472          15653#
G26     057536          15654#
G27     057570          15654#
G3      056672          15647#
G30     057572          15654#
G31     057636          15655#
G32     057670          15655#
G33     057672          15655#
```

```
G34     057736          15656#
G35     057770          15656#
G36     057772          15656#
G37     060036          15657#
G4      056736          15648#
G40     060070          15657#
G41     060072          15657#
G42     060136          15659#
G43     060170          15658#
G44     060172          15658#
G5      056770          15648#
G6      056772          15648#
G7      057036          15649#
HADR    061112          15783* 15790   15799#
HA1R    061166          15775  15785   15793   15808#
HA1W    061116          15741  15750   15760   15784   15802#
HA2R    061176          15776  15809#
HA2W    061126          15751  15761   15803#
HA3R    061206          15777  15810#
HA3W    061136          15752  15762   15804#
HA4R    061216          15778  15811#
HA4W    061146          15753  15763   15805#
HA5R    061226          15779  15812#
HA5W    061156          15754  15764   15806#
HCLR    061074          15747  15774   15793#
HCLR1   061104          15795# 15796
HCMP    061040          15758  15760   15761   15762   15763   15764   15783#
HCMP1   061060          15787# 15789
HCMP2   061066          15788  15789#
HDAT1   061236          15742  15814#
HDAT2   061246          15815#
HDAT3   061256          15816#
HDAT4   061266          15817#
HDAT5   061276          15818#
HDONE   061306          15768  15820#
HERE  - 000000          10506# 10525* 10530* 10569
HFLAG   061114          15740* 15766  15770* 15800#
HHBBF0  111226          21593  21605  21620  21621#
HHBBF1  111236          21599  21610  21625#
HHBDON  111250          21614  21630#
HHBTP1  111206          21592  21617#
HHB10   111246          21600  21608  21611  21613  21629#
HHB2    111146          21602#
HHCDON  115634          22654  22662#
HHCTP1  115612          22642  22643* 22659#
HHCTP2  115622          22641  22650  22660#
HHC10   115632          22645  22649  22651  22653  22661#
HHC2    115560          22646#
HHDAT0  071366          17633  17655  17676  17694  17712  17722#
HHDONE  071536          17721  17774#
HHHDON  102772          19977  20055#
HHH1    101562          19849#
HHH10   102332          19937#
HHH11   102402          19948#
HHH12   102452          19959#
HHH13   102522          19970#
```

```
HHH2    101632          19858#
HHH3    101702          19868#
HHH4    101752          19878#
HHH5    102022          19888#
HHH6    102072          19899#
HHH7    102142          19908#
HHH8    102212          19917#
HHH9    102262          19926#
HHP0    071376          17628   17726#
HHP1    071406          17630   17730#
HHP10   071516          17696   17766#
HHP11   071526          17714   17770#
HHP2    071416          17635   17734#
HHP3    071426          17738#
HHP4    071436          17657   17742#
HHP5    071446          17650   17707   17709   17746#
HHP6    071456          17652   17750#
HHP7    071466          17678   17754#
HHP8    071476          17671   17689   17691   17758#
HHP9    071506          17673   17762#
HH10    071116          17660   17661#
HH11    071132          17664   17668#
HH12    071152          17674#
HH13    071174          17680#  17682
HH16    071202          17681   17682#
HH17    071216          17685   17687#
HH18    071236          17692#
HH19    071260          17698#  17700
HH2     071006          17631#
HH20    071266          17699   17700#
HH21    071302          17703   17705#
HH22    071322          17710#
HH23    071344          17716#  17718
HH24    071352          17717   17718#
HH3     071030          17637#  17639
HH6     071036          17638   17639#
HH7     071046          17641   17647#
HH8     071066          17653#
HH9     071110          17659#  17661
HICORE  023604          10829#  10844
HLT   - 000000           6375#
HSTD    060762          15756   15760   15761   15762   15763   15764   15774#
HXDAT0  073272          18005   18030   18060   18076   18099   18119   18131#
HXDONE  073402          18129   18171#
HXER9   072776          18045   18052#
HXP1    073302          17997   18019   18090   18094   18102   18116   18121   18136#
HXP2    073312          17999   18001   18022   18025   18057   18141#
HXP3    073322          18145#
HXP4    073332          18073   18149#
HXP5    073342          18079   18153#
HXP6    073352          18055   18071   18114   18157#
HXP7    073362          18007   18062   18162#
HXP8    073372          18033   18166#
HX10    072674          18026   18028#
HX11    072716          18035#  18037
HX14    072724          18036   18037#
```

```
HX15    072740          18041   18043#
HX16    072756          18047#
HX165   072762          18048#
HX17    073000          19052   18053#
HX18    073020          18058#
HX19    073040          18064#  18066
HX2     072564          18000#
HX20    073046          18065   18066#
HX22    073070          18074#
HX23    073110          18081#  18083
HX26    073116          18082   18083#
HX28    073142          18095#
HX29    073164          18104#  18106
HX3     072576          18002   18003#
HX30    073172          18105   18106#
HX31    073206          18110   18112#
HX32    073226          18117#
HX33    073250          18123#  18125
HX34    073256          18124   18125#
HX4     072616          18009#  18011
HX7     072624          18010   18011#
HX8     072640          18014   18016#
HX9     072662          18024#
H1      060424          15740#
H10     060666          15763#
H11     060720          15764#
H12     060752          15767   15770#
H2      060444          15744#  15745
H3      060454          15749#  15771
H4      060526          15756#
H5      060550          15760#
H6      060602          15761#
H7      060634          15762#
IDATI0  056274          15443   15449   15464   15473   15478   15503   15525#
IDATI1  056276          15526#
IDATI2  056300          15500*  15527#
IDATI3  056302          15501*  15528#
IDATO0  056264          15460   15463   15497   15502   15520#
IDATO1  056266          15521#
IDATO2  056270          15522#
IDATO3  056272          15523#
IDONE   056304          15508   15530#
IIB8F0  111374          21634   21646   21661   21662#
IIB8F1  111404          21640   21651   21666#
IIBDON  111416          21655   21672#
IIBTP1  111354          21633   21658#
IIB10   111414          21641   21649   21652   21654   21671#
IIB2    111314          21643#
IICDON  115700          22686   22688#
IIC2    115656          22679#
IIC20   115676          22674   22686#
IIIDON  077274          19131   19208#
III1    076776          19095#
III2    077026          19105#
III3    077056          19114#
III4    077106          19123#
```

J5



```
IJMP    013264            9025    9028#
IJMP4   013102            8992    8995#   9002
IJMP5   013234            9018    9021#
ILLA  = 004700            6392#   10691
ILLB  = 000100            6393#   10694
INC1    014156            9241    9242    9243#
INC2    014160            9243    9244#
INC3    014202            9248    9249    9250    9251#
INC4    014204            9251    9253#
INC5    014220            9256    9257    9258#
INST    025612            11102*  11103*  11107   11115   11120   11165#
IOHAND  132276            25045   25053   25063#
IPAT10  056244            15444   15472   15510#
IPAT11  056246            15511#
IPAT12  056250            15512#
IPAT13  056252            15513#
IPAT20  056254            15452   15481   15515#
IPAT21  056256            15516#
IPAT22  056260            15517#
IPAT23  056262            15518#
ITRAP5= 000004             6377#
I1      056006            15441#
I105    056106            15467   15468#
I12     056110            15472#
I13     056124            15475#  15476
I14     056152            15487#
I15     056154            15488#
I16     056156            15489#
I17     056172            15493   15494#
I2      056024            15446#  15447
I20     056232            15505#  15507
I23     056240            15506   15507#
I3      056050            15456#
I4      056052            15457#
I5      056054            15458#
I6      056100            15466#  15468
JBUF0   061720            15941   15954#
JBUF1   061722            15955#
JBUF2   061724            15956#
JBUF3   061726            15957#
JDATI0  061730            15931   15943   15950   15959#
JDATI1  061732            15960#
JDATI2  061734            15961#
JDATI3  061736            15962#
JDATO0  061740            15938   15944   15964#
JDATO   061750            15928   15969#
JDATO1  061742            15965#
JDATO2  061744            15966#
JDATO3  061746            15967#
JDAT1   061752            15970#
JDAT2   061754            15971#
JDAT3   061756            15972#
JDONE   061760            15952   15975#
JJBBF0  111536            21676   21682   21688   21693   21704#
JJBDON  111550            21697   21709#
JJBTP1  111524            21675   21700#
```

```
JJB10   111546          21683   21691   21694   21696   21708#
JJB2    111462          21685#
JJJDON  077656          19266   19341#
JJJ1    077300          19224#
JJJ2    077344          19235#
JJJ3    077410          19246#
JJJ4    077454          19257#
JMPCK   013266           9028    9030#
JMPERR  020452          10238   10239   10240   10241#
JMPSEQ  013300           8985*   8990    8993*   8997    8999*   9004    9007*   9010    9013*   9016    9019*   9023    9026*
                         9030    9032#
JMPT    020442          10237   10238#
JMP2    013104           8986    8997#
JMP2A   013114           8998    8999#
JMP3    013046           8988#   9000
JMP3A   013056           8989    8990#
JMP3B   013070           8991    8992#
JMP4    013126           8995    9002#
JMP4A   013136           9003    9004#
JMP4B   013150           9005    9006#
JMP5    013210           9006    9016#
JMP5A   013222           9017    9018#
JMP6    013162           9010#   9021
JMP6A   013174           9011    9012#
JMP7    013236           9012    9023#
JMP7A   013250           9024    9025#
JSRCK   013650           9125    9128#
JSRCKA  013644           9122    9125#
JSRCK1  013666           9052    9129    9131#
JSRSEQ  013646           9056*   9064    9074*   9077    9087*   9091    9095*   9099    9105*   9107    9113*   9117    9121*
                         9126#   9128
JSR0    013304           9052#
JSR1    013310           9051    9054#
JSR1A   013332           9062#   9079
JSR2    013406           9055    9077#   9093
JSR2A   013436           9078    9080    9082    9084#
JSR2B   013440           9084    9085#
JSR3    013334           9064#   9072    9088
JSR3A   013374           9065    9067    9069    9071    9073#
JSR3B   013376           9073    9074#
JSR4    013462           9066    9075    9091#
JSR4A   013500           9092    9094#
JSR4B   013502           9094    9095#
JSR5    013552           9096    9107#   9111    9119
JSR5A   013576           9108    9110    9112#
JSR5B   013600           9112    9113#
JSR6    013514           9099#   9109    9124
JSR6A   013540           9100    9102    9104#
JSR6AD  013642           9103    9114    9124#   9130
JSR6B   013542           9104    9105#
JSR7    013612           9101    9106    9117#
JSR7A   013630           9118    9120#
JSR7B   013632           9120    9121#
J1      061620          15925#
J2      061636          15934#
J3      061640          15935#
```

```
J4       061642          15936#
J5       061676          15946#  15948
J6       061704          15947   15948#
J7       061716          15951   15952#
KBUF0    062074          15990   15999   16008   16017#
KBUF1    062076          16018#
KBUF2    062100          16019#
KBUF3    062102          16020#
KDATI0   062064          16001   16012#
KDATI1   062066          16013#
KDATI2   062070          16014#
KDATI3   062072          16015#
KDATO0   062104          15996   16002   16022#
KDATO1   062106          16023#
KDATO2   062110          16024#
KDATO3   062112          16025#
KDONE    062124          16010   16032#
KERSTK= 001000            6400#  12371   12375   12707   12714   12721   12728   13046   13239   13259   13338   13358   13448
                         13502   13507   13565   13582   13635   13690   13779   14044   14139   14170   14187
KIPAR0= 172340          120810  12417   12546   12623   12713   12759   12834   13745*  14244   14280
KIPAR1= 172342          120810  13746*
KIPAR2= 172344          120810  13747*
KIPAR3= 172346          120810  13011   13082*  13436*  13748*
KIPAR4= 172350          120810  12847*  12848*  12849*  12855*  12877*  12909*  12910*  12911*  12912*  12942*  12987*  12994*
                        13083*  13437*  13749*  13768*  13884*  14030*  14384*  14396*  14419*  14747*  14765*
KIPAR5= 172352          120810  12943*  12988*  12995*  13482*
KIPAR6= 172354          120810  13055*
KIPAR7= 172356          120810  13750*  14372*
KIPDR0= 172300          120810  12423   12583   12662   12705   12749   12841   13011   13754   14240   14268
KIPDR1= 172302          120810
KIPDR2= 172304          120810
KIPDR3= 172306          120810  13086*  13233*  13438*
KIPDR4= 172310          120810  12985*  12992*  13093*  13153*  13242*  13253*  13341*  13352*  13439*  13463*  13483*  13545*
                        13767*  13773*  13882*  13906*  13908*  13921*  13923*  13934*  13936*  13949*  13951*  13965*  13967*
                        13980*  13982*  13997*  13999*  14029*
KIPDR5= 172312          120810  12986*  12993*  13234*  13484*  13546*
KIPDR6= 172314          120810  13043*  13047   13054*
KIPDR7= 172316          120810  13034*  13037   13626*  13653*
KKB8F0   111676         21713   21725   21740   21741#
KKB8F1   111706         21719   21730   21745#
KKBDON   111720         21734   21750#
KKBTP1   111656         21712   21737#
KKB10    111716         21720   21728   21731   21733   21749#
KKB2     111614         21722#
KKCDON   116776         22899   22961#
KKC1     116002         22735#
KKC10    116234         22796#
KKC11    116262         22804#
KKC12    116310         22812#
KKC13    116336         22820#
KKC14    116364         22828#
KKC15    116412         22836#
KKC16    116440         22844#
KKC17    116466         22852#
KKC2     116030         22745#
KKC20    116514         22860#
```

M5



```
KKC21   116542          22868#
KKC22   116570          22876#
KKC23   116616          22884#
KKC24   116644          22892#
KKC3    116056          22755#
KKC4    116104          22764#
KKC5    116132          22772#
KKC6    116160          22780#
KKC7    116206          22788#
KKKDON  100210          19398   19477#
KKK1    077662          19360#
KKK3    077712          19370#
KKK4    077742          19380#
KKK5    077772          19390#
KPAT0   062114          15987   16027#
KPAT1   062116          16028#
KPAT2   062120          16029#
KWSTRT  125766          24191   24218   24365   24380#
KX2     062002          15992#
KX3     062004          15993#
KX4     062006          15994#
KX5     062042          16004#  16006
KX6     062050          16005   16006#
KX7     062062          16009   16010#
K1      021552          10508#  10574*  10575   10578   10582*  10583   10586   10589*  10591   10597*  10599   10605*  10607
                        10610
K10     021570          10515#  10619   10622   10624   10627   10630   10633
K11     021572          10516#
K12     021574          10517#
K2      021554          10509#
K3      021556          10510#
K4      021560          10511#
K5      021562          10512#  10573*  10576   10581*  10590*  10592   10594   10598*  10600   10602   10606*  10608
K6      021564          10513#  10584   10622
K7      021566          10514#  10613   10616   10619   10624   10633
LATCNT  051766          14516#  14556*  14557*  14745   14757
LDATI0  062300          16048   16055   16072   16091#
LDATI1  062302          16092#
LDATI2  062304          16068   16070*  16093#
LDATI3  062306          16071*  16094#
LDATO0  062312          16065   16073   16096#
LDATO1  062314          16097#
LDATO2  062316          16098#
LDATO3  062320          16099#
LDCDSU  117460          22971   22979   22987   22995   23003   23011   23019   23027   23036   23044   23084#
LDCFSU  116674          22735   22745   22755   22764   22772   22780   22788   22796   22804   22812   22820   22828   22836
                        22844   22852   22860   22868   22876   22884   22892   22932#
LDCT    116766          22940   22945   22959#  23091   23097
LDONE   062322          16079   16101#
LDXSUB  120720          23124   23134   23144   23154   23165   23175   23186   23196   23206   23216   23226   23236   23247
                        23258   23269   23280   23324#
LDXT    121046          23340   23342   23360#
LKS     124630          24184#  24391   24399   24401*  24402   24404*  24405   24412*  24414   24426*  24438*  24445*  24451*
                        24468*  24481*  24493*  24500*  25038*  25068*
LKSTST  126020          24387#
LLBBF0  112024          21753   21761*  21764   21778#
```

```
LLBDON  112036          21771   21784#
LLBTP1  112014          21752   21774#
LLB10   112034          21759   21767   21770   21783#
LLB2    111760          21761#
LLCDON  117552          23051   23110#
LLC1    117002          22971#
LLC10   117324          23027#
LLC11   117362          23036#
LLC12   117420          23044#
LLC2    117040          22979#
LLC3    117076          22987#
LLC4    117134          22995#
LLC5    117172          23003#
LLC6    117230          23011#
LLC7    117266          23019#
LLLDON  100622          19536   19617#
LLL1    100214          19492#
LLL2    100260          19503#
LLL3    100324          19515#
LLL4    100370          19526#
LPAT10  062256          16045   16081#
LPAT11  062260          16082#
LPAT12  062262          16083#
LPAT13  062264          16084#
LPAT20  062266          16049   16086#
LPAT21  062270          16087#
LPAT22  062272          16088#
LPAT23  062274          16089#
LSREG = 177524           6396#   6514*  14805*  14811
LTCHLT  126664           6523*  24511   24514#
LTCMSG  126666          24512   24515#
L1      062130          16042#
L2      062170          16059#
L3      062172          16060#
L4      062174          16062#
L5      062244          16076#  16078
L6      062252          16077   16078#
MBDM2A  010654           8367    8368    8369#
MBDM2B  010656           8369    8370#
MBDM2C  010666           8371    8372#
MBDM2D  010700           8373    8374    8375#
MBDM2E  010702           8375    8376#
MBDM2F  010712           8377    8378#
MBDM4A  011122           8439    8440#
MBDM4B  011132           8441    8442#
MBDM4C  011144           8443    8444    8445#
MBDM4D  011146           8445    8446#
MBDM4E  011154           8447    8448#
MDAT00  062450          16130   16133   16152#
MDAT01  062452          16153#
MDAT02  062454          16154#
MDAT03  062456          16155#
MDM1A   010564           8331    8332    8333#
MDM1B   010566           8333    8334#
MDM2A   010614           8349    8350    8351#
MDM2B   010616           8351    8352#
```

```
MDM2C   010624          8354#
MDM2D   010626          8354    8355#
MDM3A   010750          8391    8392    8393#
MDM3B   010752          8393    8394#
MDM3C   010762          8395    8396#
MDM3D   010774          8397    8398#
MDM3E   011014          8400    8401#
MDM4A   011056          8417    8418    8419#
MDM4B   011060          8419    8420#
MDM4C   011066          8421    8422#
MDM5A   011206          8465    8466    8467#
MDM5B   011210          8467    8468#
MDM5C   011220          8469    8470#
MDM5D   011230          8471    8472#
MDM5E   011250          8475    8476#
MDM6A   011302          8492    8493    8494#
MDM6B   011304          8494    8495#
MDM6C   011314          8496    8497#
MDM6D   011326          8498    8499#
MDM6E   011350          8502    8503#
MDM7A   011406          8518    8519    8520#
MDM7B   011410          8520    8521#
MDM7C   011420          8522    8523#
MDM7D   011432          8524    8525#
MDM7E   011450          8527    8528#
MDONE   062460          16140   16157#
MEM     050750          14362#  14462*  14464   14465*  14477*  14541*  14562
MERR3   062366          16120   16129#
MFPIO   020140          10148   10151#
MFPIOA  020160          10153   10155#
MFPS1   017230           9984    9986#
MFPS2A  017302           9998    9999    10000   10001#
MFPS2B  017304          10001   10002#
MFPS2C  017316          10003   10004#
MFPS3A  017356          10012   10013   10014   10015#
MFPS3B  017360          10015   10016#
MFPS3C  017372          10017   10018#
MFPS4A  017432          10026   10027   10028   10029#
MFPS4B  017434          10029   10030#
MFPS4C  017446          10031   10032#
MFPS5A  017506          10040   10041   10042   10043#
MFPS5B  017510          10043   10044#
MFPS5C  017522          10045   10046#
MFPS6A  017564          10054   10055   10056   10057#
MFPS6B  017566          10057   10058#
MFPS6C  017600          10059   10060#
MFPS7A  017642          10068   10069   10070   10071#
MFPS7B  017644          10071   10072#
MFPS7C  017656          10073   10074#
MFPT  = 000007          10410#  10412
MMBBF0  112152          21788   21799   21812   21813#
MMBBF1  112162          21796*  21817#
MMBDON  112174          21806   21823#
MMBTP1  112132          21787   21809#
MMB10   112172          21794   21802   21805   21822#
MMB2    112076          21796#
```

```
MMCDON  121056          23289   23362#
MMC1    117556          23124#
MMC10   120170          23196#
MMC11   120236          23206#
MMC12   120304          23216#
MMC13   120352          23226#
MMC14   120420          23236#
MMC15   120466          23247#
MMC16   120534          23258#
MMC17   120602          23269#
MMC2    117624          23134#
MMC20   120650          23280#
MMC3    117672          23144#
MMC4    117740          23154#
MMC5    120006          23165#
MMC6    120054          23175#
MMC7    120122          23186#
MMMDON  103432          20122   20205#
MMM1    102776          20070#
MMM2    103032          20081#
MMM3    103066          20092#
MMM4    103122          20102#
MMM5    103156          20113#
MMUHLT  050706           6519*  14344   14347#
MMUMSG  050710          14345   14349#
MMUTST  037604          12072   12277#
MMVEC = 000250          12081#  12278*  12279*  12956*  12998*  13092*  13134*  13152*  13188*  13238*  13251*  13283*  13337*
                        13350*  13382*  13442*  13464*  13499*  13505*  13547*  13772*  13866*  13886*  14006*  14037*  14125*
MODDD0  104116          20316   20320   20345#
MODDD1  104126          20318   20327   20347#
MODDOV  103722          20220   20231   20242   20253   20298#
MODDSU  102576          19849   19858   19868   19878   19888   19899   19908   19917   19926   19937   19948   19959   19970
                        20012#
MODDT0  102752          20029   20033   20051#
MODDT1  102762          20031   20040   20053#
MODFD0  103412          20176   20180   20201#
MODFD1  103422          20178   20185   20203#
MODFOV  103216          20070   20081   20092   20102   20113   20158#
MODFSU  101352          19633   19642   19651   19660   19669   19678   19687   19696   19705   19714   19723   19732   19741
                        19750   19793#
MODFT0  101526          19810   19814   19828#
MODFT1  101536          19812   19819   19830#
MODP1   101546          19798   19832#  20017   20163   20303
MOR0    020646          10287   10299#
MOR1    020670          10305   10307#
MOR2    020724          10318   10320#
MOR3    020772          10333   10335#
MOR4    021004          10336   10338#
MOR5    021016          10340   10342#
MOR6    021062          10353   10355#
MOR7    021074          10357   10360#
MOR8    021106          10361   10363#
MOV1    013736           9169    9170   9171#
MOV2    013740           9171    9173#
MOV3    013756           9176    9177   9178#
MPAT10  062430          16116   16142#
```

```
MPAT11  062432          16143#
MPAT12  062434          16144#
MPAT13  062436          16145#
MPAT20  062440          16147#
MPAT21  062442          16148#
MPAT22  062444          16149#
MPAT23  062446          16150#
MRK1    016544           9863   9868#
MRK2    016566           9868   9869    9870    9872    9874#
MRK3    016570           9874   9875#
MRK4    016612           9878   9880#
MRK5    016616           9879   9882#
MRK6    016632           9883   9885#
MTFLAG  000172          64540  6506*  25153
MTPIO   020226          10165  10168#
MTPS1   016656           9901   9902#
MTPS1A  016676           9906   9907    9908    9909#
MTPS2   016726           9917   9918#
MTPS3   016772           9928   9929#
MTPS4   017034           9938   9939#
MTPS5   017070           9948   9949#
MTPS6   017134           9958   9959#
MTPS7   017200           9968   9969#
MULDSU  076650          18988  18996  19004  19012  19047#
MULDT   076762          19065  19069  19079#
MULFSU  076322          18815  18823  18831  18839  18847  18855  18863  18871  18879  18887  18895  18903  18911
                        18946#
MULFT   076430          18962  18972#
MVLCNT  051770          14517# 14558* 14559*
M1      062326          16113#
M15     062346          16122#
M2      062352          16125#
M3      062354          16126#
M4      062356          16127#
M5      062412          16135  16136#
M6      062416          16137# 16139
M7      062424          16138  16139#
M8      062370          16129  16130#
N     = 000307          11175# 11869# 11870# 11871# 11872# 11873# 11874# 11875# 11876# 11882# 11883# 11884# 11885#
                        11886# 11887# 11888# 11889# 11890# 11891# 11988# 11989# 11990# 11991# 11992# 11993# 11994#
                        11995# 11996# 11997# 11998# 11999# 12000# 12001# 12002# 12003# 12004# 12005# 12010# 12011#
                        12012# 12013# 12014# 12015# 12016# 12017# 12018# 12019# 12030# 12031# 12032# 12033# 12034#
                        12035# 12036# 12037# 12038# 12039# 12040# 12041# 12042# 12043# 12044# 12045# 12046# 12051#
                        12052# 12053# 12054# 12055# 12056# 12057# 12058# 12059# 12060#
NATBF1  114752          22447  22453  22470  22486#
NATINS  114656          22457  22460#
NATSUB  114602          22236  22255  22274  22293  22312  22331  22350  22369  22388  22444#
NBR     001754           6562*  6565*  6574#
NDATI0  062624          16189  16202  16212#
NDATI1  062626          16213#
NDATI2  062630          16214#
NDATI3  062632          16215#
NDAT00  062562          16183  16188  16196#
NDAT01  062564          16197#
NDAT02  062566          16198#
NDAT03  062570          16199#
```

```
NDONE   062634          16194  16217#
NEGAT   026760          11382  11413   11444   11475   11508   11540   11544#
NEG00   005040           7198   7199    7200    7201#
NEG01   005042           7201   7203#
NEG02   005050           7204   7206#
NEG03   005064           7208   7209    7210    7211#
NEG04   005066           7211   7212#
NEG1    014672           9418   9419    9420    9421#
NEG10   005114           7219   7220    7221    7222#
NEG11   005116           7222   7224#
NEG12   005126           7225   7226#
NEG13   005142           7228   7229    7230    7231#
NEG14   005144           7231   7232#
NEG2    014674           9421   9422#
NEG20   005174           7239   7240    7241    7242#
NEG21   005176           7242   7243#
NEG22   005216           7249   7250#
NEG3    014716           9426   9427    9428    9429#
NEG30   005426           7354   7355    7356    7357#
NEG31   005430           7357   7358#
NEG32   005436           7359   7360#
NEG33   005454           7363   7364#
NEG34   005462           7365   7366#
NEG4    014720           9429   9430#
NEG40   005716           7472   7473    7474    7475#
NEG41   005720           7475   7476#
NEG42   005726           7477   7478#
NEG5    014740           9434   9435    9436    9437#
NEG50   005764           7489   7490    7491    7492#
NEG51   005766           7492   7493#
NEG52   005774           7494   7495#
NEG60   006034           7506   7507    7508    7509#
NEG61   006036           7509   7510#
NEG70   006070           7518   7519    7520    7521#
NEG71   006072           7521   7522#
NERRO   062554          16176  16192#
NEXT    051174          14378  14413#
NNBBFO  112306          21841  21843   21862#
NNBDON  112320          21851  21868#
NNBTP1  112266          21834  21854#
NNBTP2  112276          21844  21858#
NNB10   112316          21847  21850   21867#
NNB2    112214          21836#
NNCDON  121170          23391  23398#
NNCTBO  121152          23375  23394#
NNCTB1  121156          23382  23385   23387   23389   23395#
NNC10   121166          23380  23386   23388   23390   23396#
NNC2    121120          23384#
NNNDON  104136          20263  20349#
NNN1    103436          20220#
NNN2    103512          20231#
NNN3    103566          20242#
NNN4    103642          20253#
NOABRT  133566          25172#
NODL    024000          10871  10874#
NODL2   024426          10936#
```

F6



```
NOP   = 000240          6360   6388#
NOR     023600         10827#  10830
NOSUB   023674         10846   10849#
NPAT10  062614         16171   16207#
NPAT11  062616         16208#
NPAT12  062620         16209#
NPAT13  062622         16210#
NPAT20  062602         16174   16186   16202#
NPAT21  062604         16203#
NPAT22  062606         16204#
NPAT23  062610         16205#
NXTST   024704         10955   10987#
N1      062464         16168#
N12     062534         16187   16188#
N13     062550         16191#  16193
N14     062556         16192   16193#
N2      062510         16179#
N3      062512         16180#
N4      062514         16181#
ODATI0  063000         16250   16263   16274#
ODATI1  063002         16275#
ODATI2  063004         16276#
ODATI3  063006         16277#
ODATO0  062736         16244   16249   16257#
ODATO1  062740         16258#
ODATO2  062742         16259#
ODATO3  062744         16260#
ODONE   063010         16255   16279#
OERR0   062706         16236   16248#
OOBDON  112450         21893   21906#
OOBTP1  112426         21871   21876   21883   21889   21896#
OOBTP2  112436         21872   21884   21900#
OOB10   112446         21887   21890   21892   21905#
OOB2    112362         21881#
OOCDON  121302         23420   23427#
OOCTB0  121264         23404   23423#
OOCTB1  121270         23411   23414   23416   23418   23424#
OOC10   121300         23409   23415   23417   23419   23425#
OOC2    121232         23413#
OODON   105130         20522   20524#
OOOT    105064         20498   20509#
OOO2    105062         20505#  20510
OOO3    105076         20513#
OOO4    105122         20520   20521#
OPAT10  062770         16231   16269#
OPAT11  062772         16270#
OPAT12  062774         16271#
OPAT13  062776         16272#
OPAT20  062754         16247   16263#
OPAT21  062756         16234   16264#
OPAT22  062760         16265#
OPAT23  062762         16266#
OPAT24  062764         16267#
OVDNTT  077646         19325   19327   19339#
OVDTT   100612         19600   19602   19615#
OVFNTT  077264         19192   19194   19206#
```

```
OVFTT   100200          19459   19461   19475#
OVUNDN  077524          19224   19235   19246   19257   19306#
OVUNDT  100440          19492   19503   19515   19526   19575#
OVUNFN  077142          19095   19105   19114   19123   19173#
OVUNFT  100026          19360   19370   19380   19390   19436#
01      062640          16228#
012     062710          16248   16249#
013     062724          16252#  16254
014     062732          16253   16254#
02      062664          16240#
03      062666          16241#
04      062670          16242#
PCN01   021122          10376#  10377
PCN1    024716          10989#  10991
PCN2    021140          10382#  10383
PCN3    021170          10389#  10390
PCN4    021214          10395#  10396
PCN5    021236          10401#  10402
PCR   = 177520           6395#  14937*  14961*
PDATI0  063112          16295   16303   16308   16316#
PDATI1  063114          16317#
PDATI2  063116          16318#
PDATI3  063120          16319#
PDATO0  063122          16300   16304   16321#
PDATO1  063124          16322#
PDATO2  063126          16323#
PDATO3  063130          16324#
PDONE   063132          16310   16326#
PDRTB1  044356          13236   13297#
PDRTB2  044410          13250   13314#
PDRTB3  044642          13335   13396#
PDRTB4  044674          13349   13413#
PNTR    054400          14975#
PPAT10  063102          16292   16311#
PPAT11  063104          16312#
PPAT12  063106          16313#
PPAT13  063110          16314#
PPBDON  112600          21932   21945#
PPBTP1  112556          21909   21914   21922   21928   21935#
PPBTP2  112566          21910   21923   21939#
PPB10   112576          21926   21929   21931   21944#
PPB2    112512          21919#
PPCDON  121414          23449   23455#
PPCTB0  121376          23433   23452#
PPCTB1  121402          23440   23443   23445   23447   23453#
PPC10   121412          23438   23444   23446   23448   23454#
PPC2    121344          23442#
PPPBF0  105250          20538   20570#
PPPBF1  105264          20549   20555   20558   20572#
PPPDON  105312          20568   20578#
PPPTP1  105300          20545   20559   20574#
PPP10   105310          20561   20563   20565   20567   20577#
PPP2    105150          20540#  20541
PPP3    105200          20553#
PPP4    105212          20556   20558#
PS    = 177776           6363#   6947*   6948    6952*   6953    6957*   6958    6962*   6963    9886#   9900    9914*   9916
```

```
                    9925*  9927   9935*  9937   9947   9955*  9957   9965*  9967   9981*  9988*  9996*  10010*
                   10024*  10038* 10052* 10066* 10087* 10089  10099* 10105* 10107* 10118* 10122* 10124* 10128*
                   10130*  10134* 10144* 10147  10149* 10151* 10158* 10160* 10164  10166* 10168* 10219* 10220*
                   10221   10224* 10226* 10263  10278  10431* 10434* 10439* 10445* 12084  24086* 24983* 25020*
                   25040*
PSW    = 177776    12084#  12319* 12320* 12321  12330* 12339* 12341* 12342  12350* 12352* 12353  12370* 12372*
                   12374*  12497* 12506* 12851  12873  12939* 12963  12979  12989* 12991* 13017* 13019* 13096
                   13125*  13131* 13133* 13156  13185* 13187* 13564* 13569* 13581* 13584* 13586* 13599* 13603
                   13605*  13680* 13689* 13691* 13693* 13776* 13790* 13800* 13810* 13819* 13831* 13843* 13854*
                   13889*  13899* 13917* 13931* 13945* 13960* 13976* 13991* 14028* 14039* 14051* 14062* 14072*
                   14081*  14092* 14104* 14115* 14126* 14136* 14169* 14202  14204  15148
PSWORD 025626      11179#  11180  11362* 11366  11394* 11398  11425* 11429  11456* 11460  11488* 11492  11521*
                   11525   11613* 11614* 11615* 11616* 11617* 11618* 11619* 11620* 11621* 11622* 11660* 11664
                   11699*  11703  11738* 11742  11869* 11870* 11871* 11872* 11873* 11874* 11875* 11876* 11882*
                   11883*  11884* 11885* 11886* 11887* 11888* 11889* 11890* 11891* 11988* 11989* 11990* 11991*
                   11992*  11993* 11994* 11995* 11996* 11997* 11998* 11999* 12000* 12001* 12002* 12003* 12004*
                   12005*  12010* 12011* 12012* 12013* 12014* 12015* 12016* 12017* 12018* 12019* 12030* 12031*
                   12032*  12033* 12034* 12035* 12036* 12037* 12038* 12039* 12040* 12041* 12042* 12043* 12044*
                   12045*  12046* 12051* 12052* 12053* 12054* 12055* 12056* 12057* 12058* 12059* 12060*
PUSRM  = 030000     6367#
PWRDN    133314    6501   25147# 25151
PWRMSG   133570   25155   25178#
PWRUP    133330   25147   25151#
P1       063014   16289#
P2       063030   16297#  16298
P3     = 063032   16298#
P4       063034   16300#
P5       063056   16305#  16307
QDATI0   063254   16353   16367  16377#
QDATI1   063256   16378#
QDATI2   063260   16379#
QDATI3   063262   16380#
QDATO0   063244   16348   16352  16372#
QDATO1   063246   16373#
QDATO2   063250   16374#
QDATO3   063252   16375#
QDONE    063264   16360   16382#
QPAT10   063224   16340   16362#
QPAT11   063226   16363#
QPAT12   063230   16364#
QPAT13   063232   16365#
QPAT20   063234   16343   16358  16367#
QPAT21   063236   16368#
QPAT22   063240   16369#
QPAT23   063242   16370#
QQBDON   112742   21970   21984#
QQBTP1   112710   21947   21952  21960  21966  21973#
QQBTP2   112730   21948   21961  21978#
QQB10    112740   21964   21967  21969  21983#
QQB2     112644   21957#
QQCDON   121536   23478   23485#
QQCTB0   121514   23461   23481#
QQCTB1   121520   23469   23474  23476  23482#
QQCTB2   121530   23468   23472  23476  23483#
QQC10    121534   23466   23473  23475  2347?  23484#
QQC2     121462   23471#
```

```
Q0QBF0  105432          20593   20622#  20633
Q0QBF1  105446          20603   20608   20611   20623#  20642   20645   20647
Q0QDON  105600          20653   20656#
Q0QTP1  105462          20600   20612   20624#  20640   20648
Q0Q10   105472          20614   20616   20618   20620   20629#
Q0Q2    105332          20595#  20596
Q0Q20   105474          20621   20633#
Q0Q22   105506          20636#  20637
Q0Q23   105540          20643   20644#
Q0Q24   105552          20646   20647#
Q0Q25   105576          20651   20655#
Q0Q3    105362          20606#
Q0Q4    105374          20610   20611#
Q1      063136          16337#
Q2      063152          16345#  16346
Q22HLT  053442           6521*  14782   14785#
Q22MSG  053444          14783   14786#
Q22TST  051774          14445   14482   14532#
Q22TS1  052704          14661   14672#
Q22TS2  053012          14686   14701#
Q22TS3  053156          14725   14740#
Q3    - 063154          16346#
Q4      063156          16348#
Q5      063200          16354#  16356
RBUF    124612          24175#  24258   24364
RBUF2   126722          24521#  24596   24749   24771   24778   24800   24819   24843   24855   24889   24891   24896   24904
                        24912   24924   24926   24928   24939   25056   25061
RCSR    124610          24174#  24250   24269   24271*  24272   24274*  24275   24278*  24280
RCSR2   126720          24520#  24588   24649   24651*  24652   24654*  24655   24662*  24664   24755   24766   24776   24779
                        24786*  24791   24795*  24798*  24809   24811*  24814*  24830   24832*  24839*  24851*  24853   24858*
                        24869*  24871   24874*  24877*  24902   24907   24921   24935   25036*
RECCT2  132454          25027*  25055*  25077   25094#
REC2    132254          25031   25055#
REGR1   026074          11381   11384#
REGR2   026222          11412   11415#
REGR23  030316          11684   11687#
REGR3   026350          11443   11446#
REGR4   026474          11474   11477#
REGR5   026624          11507   11510#
REG01   030136          11647#  11765
REG1    003672           6809#   6812
REG1A   003720           6819#   6822
REG1E   003704           6810    6813#
REG2    003752           6829#   6832
REG2B   004000           6839#   6842
REG2E   003764           6830    6833#
REG3    004032           6850#   6853
REG3A   004060           6860#   6863
REG3E   004044           6851    6854#
REG4    004112           6871#   6874
REG4A   004140           6881#   6884
REG4E   004124           6872    6875#
REG45   030472          11722   11725#
REG5    004172           6893#   6896
REG5A   004220           6903#   6906
REG5E   004204           6894    6897#
```

J6



```
REG6    004252            6914#  6917
REG6A   004300            6924#  6927
REG6E   004264            6915   6918#
RESET2  024476           10940  10947#
RESET3  024474           10946#
REST    017726           10091#
RESTRT  001404            6501#  25138  25157
RET     025424           11104  11112#
RETA    050644           14330  14332#
RETAT   023156           10714  10720#
RETBT   023206           10723  10729#
RETCT   023242           10733  10738#
RETR1   024724           10988  10991#
RETR2   024762           10999  11003#
RETR3   025006           11011  11015#
RET1    025434           11113  11115#
RET2    025446           11116  11118#
RET3    025456           11119  11120#
RET4    025454           11119#
RFLAG   054124           14902#  14979*
RITSH   030656           11685  11723  11763  11766#
ROL1    015346            9565   9566   9567   9568#
ROL2    015350            9568   9569#
ROL3    015366            9572   9573   9574   9575#
ROL4    015370            9575   9576#
ROL5    015404            9579   9580   9581#
ROL6    015406            9581   9582#
ROL7    015430            9585   9586   9587   9589#
RORB1   016006            9692   9693#
RORB10  016070            9712   9713   9714#
RORB11  016072            9714   9715#
RORB12  016104            9716   9717#
RORB13  016116            9718   9719   9720   9721#
RORB14  016120            9721   9722#
RORB2   016020            9694   9695   9696   9697#
RORB3   016022            9697   9698#
RORB4   016032            9699   9700   9701#
RORB5   016034            9701   9702#
RORB7   016044            9703   9704#
ROR1    015454            9595   9596   9597   9598#
ROR2    015456            9598   9599#
ROR3    015474            9602   9603   9604   9605#
ROR4    015476            9605   9606#
ROR5    015514            9609   9610   9611   9612#
ROR6    015516            9612   9613#
ROR7    015532            9616   9617   9618#
ROTLP1  054022           14860#  14866
ROTLP2  054064           14870#  14877
ROTX    012430            8771   8773*  8777   8781#  8791*  8793*  8795   8807*  8808   8812
ROTXAD  012520            8808*  8810*  8814#
ROTOA   011734            8628   8629   8631#
ROTOB   011736            8631   8632#
ROTOC   011760            8635   8636   8638#
ROT1A   012010            8654   8655   8657#
ROT1B   012012            8657   8658#
ROT1C   012036            8661   8662   8664#
```

```
ROT1D   012040            8664    8665#
ROT1E   012070            8670    8671    8673#
ROT2A   012124            8687    8689    8592#
ROT2B   012126            8692    8693#
ROT2C   012156            8697    8699    8701#
ROT2D   012160            8701    8702#
ROT2E   012214            8707    8709    8712#
ROT3A   012244            8725    8727#
ROT3B   012246            8727    8728#
ROT3C   012274            8731    8733#
ROT3D   012276            8733    8734#
ROT3E   012324            8737    8739#
ROT4    012362            8754    8756    8758#
ROT5    012426            8776    8778    8780#
ROT6    012460            8794    8796#
ROT7    012516            8811    8813#
RPSW    124622           24179#
RPSW2   126732           24525#  24825   24827*  24841*  25032*
RRBDON  113102           22012   22026#
RRBTP1  113056           21988   21994   21995*  22002   22015#  22023
RRBTP2  113066           21989   22003   22019#
RRBTP3  113076           21993   22008   22023#
RRB10   113100           22006   22009   22011   22025#
RRB2    113012           21999#
RRCDON  121662           23509   23516#
RRCTB0  121640           23492   23512#
RRCTB1  121644           23500   23505   23507   23513#
RRCTB2  121654           23499   23503   23507   23514#
RRC10   121660           23497   23504   23506   23508   23515#
RRC2    121606           23502#
RRRDON  105752           20697   20713#
RRREXP  105742           20688   20709#
RRRTP1  105722           20666   20699#
RRRTP2  105732           20672   20704#
RRR10   105720           20674   20692   20695   20697#
RRR2    105652           20683#
RRR3    105654           20665   20684#  20689
RRR4    105700           20691#  20693
RTCPSW  124634           24186#  24423   24425*  24453*  24462   24464*  24483*  25034*
RTCVT   124632           24185#  24396   24397*  24416*  24422   24424*  24430*  24433*  24444*  24452*  24461   24463*  24477*
                         24482*  24489   24490*  24502*  25025   25033*  25090*
RTI1    023412           10774   10778#
RTI2    023416           10775   10780#
RTRAP =  000010           6391#   6626*  11029*
RTRAP1=  000034           6382#   6627*
RTRAP2=  000020           6381#   6645*
RTRAP3=  000030           6380#   6646*
RTRAP4=  000014           6379#   6655*  10714*  10723*  10733*
RTRAP5=  000004           6378#   6662*   6663*  10988*  10999*  11011*  11027*
RTS1    013710            9144    9149#
RTT1    023302           10747   10750#
RTT2    023310           10748   10753#
RTT3    023342           10758   10762#
RTT4    023352           10759   10766#
RTT5    023314           10756#  10764   10769
RTT6    023364           10767   10770#
```

```
RVECT   124620          24178#  24266   24267*  24282*  25024   25089*
RVECT2  126730          24524#  24646   24647*  24666*  24787   24788*  24794*  24801*  24806   24807*  24815*  24820*  24824
                        24826*  24836*  24840*  24844*  24849   24850*  24860*  24867   24868*  24879*  25022   25031*  25087*
RWREG = 177522           6397#  14816*  14817   14821*  14822   14826*  14827   14831*  14832   14836*  14837   14841*  14842
                        14847*  14848   14852*  14853   14857*  14858*  14859   14860   14865*  14867*  14868*  14869   14870
                        14874*
ROTRAP  023624          10831   10835#
R1ERR   003736           6820    6824#
R2ERR   004016           6840    6844#
R3ERR   004076           6861    6865#
R4ERR   004156           6882    6886#
R5ERR   004236           6904    6908#
R6ERR   004316           6925    6929#
SADR    061570          15886*  15897   15901*  15904#
SAV30   001270          64840*  25125   25172
SAV32   001272          64840*  25126   25172
SAV42   001200           6482#   6490*
SBC1    015240           9524    9525    9526    9527#
SBC2    015242           9527    9528#
SBC3    015260           9531    9532    9533    9534#
SBC4    015262           9534    9535#
SBC5    015300           9538    9539    9540    9541#
SBC6    015302           9541    9542#
SBC7    015322           9546    9547    9548#
SB0     012534           8827    8828#
SB2     012614           8858    8859#
SB4     012676           8889    8890#
SB5     012740           8910#
SB5A    012732           8908    8909#
SB5X    012742           8905*   8907    8911#   8912
SB5XAD  012744           8904    8909    8912#
SB6     012774           8929#
SB6X    012776           8925*   8926    8930#
SB7     013026           8948#
SB7X    013030           8944*   8949#   8950
SB7XAD  013032           8945    8950#
SCOPE = 000240           6360#
SC3     020572          10259*  10277#  10284*  10285
SC4     020606          10260*  10281#  10283*
SDAT00  061604          15848   15851   15872   15875   15911#
SDAT01  061606          15912#
SDAT02  061610          15913#
SDAT03  061612          15914#
SDONE   061614          15901   15916#
SERR0   061500          15835   15884#
SERR1   061406          15840   15857#  15891   15896
SERR10  061516          15887#  15902
SERR4   061550          15863   15899#
SETBR   001634           6555#   6582
SETCC   001654           6559#   6580
SETCD   020570          10273   10276#  10286
SETREG  050346          12706   12747   14239#
SETUP   001616           6552#
SET2BR  001724           6561    6565#
SHL     003504           6741#   6744
SHLE    003520           6742    6745#
```

M6



```
SHR     003564          6758#  6761
SHRE    003600          6759   6762#
SIMGOA  051772         14518#  14560*  14561*  14679*
SKTST2  024514         10950#
SLU1ST  124636         10875   10886   15093   24127   24188#
SLU2ST  126740         24383   24503   24529#
SL1MLT  125740          6524*  24371   24374#
SL1MSG  125742         24372   24376#
SL2MLT  131534          6525*  24956   24959#
SL2MSG  131536         24957   24961#
SNMBOA  006206          7579    7580    7581#
SNMB1A  006256          7616    7617    7618#
SNMB1B  006260          7618    7619#
SNMB1C  006302          7624    7625    7626    7627#
SNMB2A  006362          7665    7666    7667#
SNMB2B  006364          7667    7668#
SNMB2C  006372          7669    7670#
SNMB2D  006412          7674    7675    7676    7677#
SNMB2E  006414          7677    7678#
SNMB3A  006514          7724    7725    7726#
SNMB3B  006516          7726    7727#
SNMB3C  006534          7730    7731    7732    7733#
SNMB3D  006536          7733    7734#
SNM0A   006164          7560    7561    7562    7563#
SNM1A   006232          7597    7598    7599    7600#
SNM2A   006326          7642    7643    7644    7645#
SNM2B   006330          7645    7646#
SNM3A   006452          7697    7698    7699    7700#
SNM3B   006454          7700    7701#
SNM4A   006570          7752    7753    7754#
SNM4B   006572          7754    7755#
SNM5A   006626          7774    7775    7776#
SNM5B   006630          7776    7777#
SNM6A   006666          7796    7797    7798#
SNM6B   006670          7798    7799#
SNM7A   006724          7818    7819    7820#
SNM7B   006726          7820    7821#
SOB1    016460          9841#   9846
SOB2    016466          9841    9842    9843#
SOB3    016470          9843    9844#
SOB4    016510          9847    9848    9849    9851#
SOPA    006120          7539    7540#
SOPB    006140          7542    7544#
SOPB0A  004514          7039    7040#
SOPB0B  004524          7041    7043#
SOPB1A  004566          7075    7076#
SOPB1B  004604          7077    7080    7082#
SOPB1C  004624          7100    7101#
SOPB1D  004644          7106    7108#
SOPB2A  004730          7154    7155#
SOPB2B  004750          7159    7162#
SOPB2C  004774          7177    7178#
SOPB2D  005020          7184    7187#
SOPB3A  005304          7301    7302#
SOPB3B  005330          7305    7307    7311#
SOPB3C  005352          7334    7335#
```

```
SOP83D  005374          7342    7344#
SOPX    006112          7538#   7540*   7541*   7545
SOPXAD  006142          7543*   7545#
SOPZA   004666          7127    7128#
SOP0A   004430          6998    6999#
SOP0B   004442          7002    7003#
SOP0C   004460          7019    7020#
SOP0D   004504          7022    7025    7029#
SOP1A   004536          7056    7057#
SOP1B   004550          7058    7061#
SOP2B   004706          7131    7135#
SOP3A   005244          7273    7274#
SOP3B   005260          7277    7279#
SOP4A   005514          7384    7385#
SOP4B   005534          7388    7392#
SOP5A   005566          7419    7420#
SOP5B   005602          7423    7425#
SOP6A   005622          7440    7441#
SOP6B   005636          7442    7444#
SOP7A   005660          7461    7462#
SOP7B   005674          7463    7465#
SPAT10  061574          15832   15852   15861   15876   15906#
SPAT11  061576          15907#
SPAT12  061600          15908#
SPAT13  061602          15909#
SR0   = 177572          12081#  12281*  12405   12449   12498*  12500   12505*  12847*  12848*  12849*  12858*  12860*  12879*
                        12881*  12909*  12910*  12911*  12912*  12958*  13000*  13035   13036*  13108   13121*  13168   13181*
                        13260   13274*  13277*  13359   13373*  13376*  13455*  13503   13508   13523*  13524   13537*  13636
                        13650*  13686   13688*  14188*  14385*  14393*  14401*  14413*  14423*  14426*  14540*  14750*  14761*
                        14771*
SR1   = 177574          12081#  12517
SR2   = 177576          12081#  12457   12475   13109   13169   13261   13360   13449   13485   13492   13504   13509   13528
                        13538   13637   13687
SR3   = 172516          12081#  12522*  12523   12529*  12530   12847*  12848*  12849*  12857*  12878*  12909*  12910*  12911*
                        12912*  12957*  14386*  14394*  14402*  14424*  14746*  14762*  14772*
SSBDON  113244          22053   22067#
SSBTP1  113220          22029   22035   22036*  22043   22056#  22064
SSBTP2  113230          22030   22044   22060#
SSBTP3  113240          22034   22049   22064#
SSB10   113242          22047   22050   22052   22066#
SSB2    113154          22040#
SSCDON  122006          23541   23547#
SSCTB0  121770          23522   23544#
SSCTB1  121774          23530   23535   23537   23539   23545#
SSC10   122004          23527   23534   23536   23538   23540   23546#
SSC2    121726          23532#
SSSA1   106114          20732   20745   20757#
SSSBF0  106104          20723   20736   20738   20753#
SSSDON  106136          20750   20769#
SSSTP1  106124          20729   20739   20763#
SSS10   106134          20731   20737   20742   20748   20768#
SSS2    106024          20734#
START   001202           6475    6484#
STATUS= 177776           6389#   6626*   6627*   6645*   6646*   6655*   6662*   6663*   10835*  10836   10860*  10861   10879*
                        10885*  10895*  10902*  10911*  10947*  10973*  10976
STBOT   001000           6400    6401    6467#   6476    6502    6509    6626    6627    6645    6646    6655    6662    6663
```

```
                    6946   9054   9068   9081   9085   9142   9860   9873   9882  10133  10135  10143  10152
                   10162  10169  10411  10713  10722  10729  10732  10738  10744  10756  10772  10786  10801
                   10807  10926  10838  10850  10863  10870  10894  10910  10939  10962  10978  10987  10991
                   11041  11054  11060  11105  11112  11115  11118  11326  12277  15089  24071  24151  25152
STCDFS  107704     21138  21157  21176  21195  21214  21264#
STCDT   110034     21270  21276  21283  21298#
STCFDS  107252     20963  20982  21001  21020  21039  21093#
STCFT   107402     21099  21105  21112  21127#
STCIBF  123370     23838  23844  23851  23868#
STCSUB  123242     23643  23652  23661  23670  23690  23689  23698  23707  23717  23726  23735  23744  23753
                   23762  23771  23781  23791  23833# 23884
STP     023114     10709  10710#
STP3    023550     10808#
STP3D   023552     10808  10809#
STP4    024666     10979#
STP4E   024670     10959  10979  10980#
STRMSG  133635      6498  25180#
STXBF   123764     23989* 23990* 23997  24000  24005  24010#
STXSUB  123652     23904  23912  23920  23929  23938  23947  23987#
SUB0    007010      7859   7860   7861   7862#
SUB0A   007012      7862   7863#
SUB1    015126      9494   9495   9496#
SUB2    015130      9496   9497#
SUB3    015152      9501   9502   9503#
SUB4    015154      9503   9504#
SUB5    015172      9508   9509   9510#
SUB6    015174      9510   9511#
SUB7    015214      9514   9515   9516   9517#
SWB1    014424      9328   9329   9330#
SWB2    014426      9330   9331#
SWB3    014444      9334   9335   9336   9337#
SWMSG   054544     15000  15003#
SWR     021576     10518# 10522* 11074* 11089* 11096  12283  14376  14443  14533  14702  14809  15091  24189
                   24381  24530  24975
SWREG   000176      6456#
SXT0    016326      9781   9782   9783   9784   9786#
SXT1    016330      9786   9787#
SXT2    016360      9793   9794   9795   9796   9798#
SX10    061442     15870#
SX11    061466     15878* 15880
SX12    061474     15879  15880#
SX2     061344     15843#
SX3     061346     15844# 15684
SX4     061352     15846#
SX5     061376     15854# 15856
SX6     061404     15855  15856#
SX7     061410     15860# 15886
SX8     061434     15867#
SX9     061436     15868# 15899
S1      025644     11191# 11192  11611  11615  11879  11882  11884
S10     025666     11200# 12049  12052
S11     025670     11201#
S2      025646     11192# 11612  11614  11880  11883
S3      025650     11193#
S4      025652     11194#
S5      025654     11195# 11196  12007  12010  12012
```

```
S6      025656          11196# 12008  12011
S7      025660          11197#
S8      025662          11198#
S9      025664          11199# 11200  12048  12051  12053
TABLE   025470          11091  11124#
TBIT    000020          12085# 14202  14207  14220
TBITPS  037570          12272# 12280# 14205# 14220  14222  14223#
TBLEND  131602          24969# 24999
TBL1    011526           8544   8545#  8546#  8547#  8548#  8549   8556#  8569#  8570#  8571#  8572#  8573   8575
                         8576   8577   8578   8579   8591   8592#  8593#  8594#  8595#  8596   8610#  8611#  8612#
                         8613#  8614
TBL2    011574           8568   8579#  8609
TBUF    124616          24177# 24208  24219# 24224# 24236# 24239# 24352# 24355#
TBUF2   126726          24523# 24549  24556# 24561# 24574# 24577# 24737# 24740#  24754#  24775#  24790#  24808#  24829#
                        24852# 24870# 24884# 24901# 24920# 24933# 25049#
TCCBF0  115756          22701  22709  22717#
TCCBF1  115766          22705  22718#
TCCDON  115776          22713  22720#
TCC2    115716          22702#
TCC3    115746          22710  22712#
TCSR    124614          24176# 24199  24220  24225  24228  24237  24242  24287#  24290  24295#  24300#  24306#  24310
                        24312# 24315# 24324# 24330  24332# 24341# 24347# 24351#  24353  24358#  25066#
TCSR2   126724          24522# 24540  24557  24562  24565  24575  24580  24606   24608#  24609  24611#  24612  24615#
                        24617  24625  24627# 24628  24630# 24631  24638# 24640   24671#  24674  24679#  24684#  24690#
                        24694  24696# 24699# 24708# 24714  24716# 24725# 24732#  24736#  24738  24743#  24785#  24885
                        24905# 24919# 24923# 25035# 25052# 25067#
TDATI0  056600          15541  15549  15564  15573  15578  15603  15626#
TDATI1  056602          15627#
TDATI2  056604          15600# 15628#
TDATI3  056606          15601# 15629#
TDATO0  056570          15560  15563  15597  15602  15621#
TDATO1  056572          15622#
TDATO2  056574          15623#
TDATO3  056576          15624#
TDEC1   022414          10639  10642#
TDEC2   022430          10646  10648#
TDEC3   022460          10654  10656#
TDEC4   022512          10660  10663#
TDEC6   022526          10666#
TDEC7   022530          10664  10667#
TDEC77  024074          10881  10889#
TDEC8   024072          10883  10888#
TDONE   056610          15608  15631#
TEMP1   025630          10882# 10892  10896# 10907  10914# 10925  10963# 10980  11181#  11182  11327#  11351  11384
                        11415  11446  11477  11511  11544# 11550# 11565# 11572#  11590#  11603#  11640#  11648  11687
                        11726  11767# 11775# 11789# 11797# 11804# 11809# 11816#  11850#  11857#
TEMP2   025632          11183# 11184  11328# 11354  11358  11378# 11387  11390   11409#  11418  11421  11440#  11449
                        11452  11471# 11481  11484  11504# 11514  11517  11537#  11545#  11551#  11557#  11561#  11566#
                        11573# 11578# 11584# 11591# 11597# 11641# 11649  11688   11727   11768#  11817#  11827#  11838#
                        11843# 11851# 11858#
TEMP3   025634          11185# 11186  11329# 11369  11379# 11401  11410# 11432   11441#  11463  11472#  11495  11505#
                        11528  11538# 11546# 11552# 11567# 11574# 11579# 11585#  11592#  11598#  11642#  11653  11656
                        11679# 11692  11695  11717# 11731  11734  11757# 11769#  11781#  11785#  11790#  11810#  11819#
                        11832# 11839# 11844# 11859#
TEMP4   025636          11187# 11188  11330# 11366  11398  11429  11460  11492   11525   11553#  11568#  11580#  11586#
                        11593# 11599# 11604# 11643# 11667  11682# 11706  11720#  11745   11760#  11770#  11776#  11791#
```

```
                    11798*  11828*  11833*  11845*  11860*
TEMP5   025640      11189#  11644*  11669   11681*  11708   11719*  11747   11759*  11771*  11792*  11799*  11805*  11820*
                    11846*  11852*
TEMP6   025642      11190#  11645*  11664   11703   11742   11777*  11793*  11800*  11811*  11821*  11834*  11853*  11861*
TENSAV  025254      11066*  11079#  11090
TESTN1  021600      10447   10520#
TEST1   014362       9312    9313    9314    9315#
TEST2   014364       9315    9316#
TEST3   014402       9320    9321    9322#
THRPRT  025676      11109   11321#
TICKER  132210      25033   25044#
TICKS   132456      25028*  25044*  25063   25095#
TKB   = 177562       6385#  10931
TMP     050744      14360#  14380*  14399   14414*  14427   14459*  14467   14478   14706*  14723   14740*  14769   14811*
                    14812
TMP1    050746      14361#  14381*  14400   14415*  14428   14707*  14724   14741*  14770
TOFF    050260      13011   13018   13032   13563   13671   14202#
TON     050314      13011   13018   13057   13588   13724   14220#
TONT1   023474      10790   10797#
TO10    021530       6432    6679   10491#  10678   11043   11061   11108   24149
TO114   021544       6445   10497#
TO14    021532       6434    6680   10492#  10690   10809   10949
TO20    021534       6436   10493#  10679
TO244   021546       6448   10498#
TO250   021550       6450   10499#  12278   12998   13134   13188   13283   13382   13464   13547   13652   13722   13866
                   14006   14125
TO30    021536       6492   10494#
TO34    021540       6439    6681   10495*  10689   10924
TO4     021526       6430    6507    6678   10490#  10691   10694   10710   10865   10874   10876   10890   11042   13056
                   13587   13651   13720   24148
TO40    021542       6442    6682   10496#
TPAT10  056550      15540   15572   15611#
TPAT11  056552      15612#
TPAT12  056554      15613#
TPAT13  056556      15614#
TPAT20  056560      15552   15581   15616#
TPAT21  056562      15617#
TPAT22  056564      15618#
TPAT23  056566      15619#
TPB   = 177566       6365#   6386#  10968*  10971*  24087*  24090*  24101*  24104*  25166*
TPS   = 177564       6364#
TPSW    124626     24181#  24326   24328*  24344*
TPSW2   126736     24527#  24710   24712*  24728*  25030*
TRACE   023470      10787   10795#
TRAPA = 000077       6390#   6626   11032
TRAPB   023746      10839   10856   10865#
TRAPC = 104777       6403#   6627
TRAPPC  037560      12272#
TRAPPS  037562      12272#
TRAP10  025256      11067   11080#
TRCSR = 177560       6384#  10932*  10934
TRC1    023540      10802   10805   10807#
TRPADR  023622      10833#  10838
TRT   = 000003       6376#   6655   10690
TR0     024204      10897   10904#
TR2     024210      10900   10906#
```

| | | | | | |
|---|---|---|---|---|---|
| TR3 | 024316 | 10913 | 10920# | | |
| TR4 | 024322 | 10915 | 10922# | | |
| TR5 | 024320 | 10917 | 10921# | | |
| TSTSPC | 037520 | 12065# | | | |
| TST160 | 030712 | 11686 | 11724 | 11764 | 11773# |
| TST161 | 030740 | 11774 | 11779# | | |
| TST162 | 030756 | 11780 | 11783# | | |
| TST163 | 030774 | 11784 | 11787# | | |
| TST164 | 031036 | 11788 | 11795# | | |
| TST165 | 031074 | 11796 | 11802# | | |
| TST166 | 031120 | 11803 | 11807# | | |
| TST167 | 031150 | 11808 | 11813# | | |
| TST170 | 031206 | 11814 | 11823# | | |
| TST171 | 031232 | 11825 | 11830# | | |
| TST172 | 031260 | 11831 | 11836# | | |
| TST173 | 031302 | 11837 | 11841# | | |
| TST174 | 031340 | 11842 | 11848# | | |
| TST175 | 031374 | 11849 | 11855# | | |
| TST176 | 031446 | 11869# | | | |
| TST177 | 031514 | 11870# | | | |
| TST200 | 031562 | 11871# | | | |
| TST201 | 031634 | 11872# | | | |
| TST202 | 031702 | 11873# | | | |
| TST203 | 031750 | 11874# | | | |
| TST204 | 032022 | 11875# | | | |
| TST205 | 032070 | 11876# | | | |
| TST206 | 032162 | 11882# | | | |
| TST207 | 032240 | 11883# | | | |
| TST210 | 032316 | 11884# | | | |
| TST211 | 032374 | 11885# | | | |
| TST212 | 032450 | 11886# | | | |
| TST213 | 032524 | 11887# | | | |
| TST214 | 032600 | 11888# | | | |
| TST215 | 032656 | 11889# | | | |
| TST216 | 032734 | 11890# | | | |
| TST217 | 033010 | 11891# | | | |
| TST220 | 033064 | 11988# | | | |
| TST221 | 033134 | 11989# | | | |
| TST222 | 033204 | 11990# | | | |
| TST223 | 033254 | 11991# | | | |
| TST224 | 033330 | 11992# | | | |
| TST225 | 033400 | 11993# | | | |
| TST226 | 033450 | 11994# | | | |
| TST227 | 033524 | 11995# | | | |
| TST230 | 033574 | 11996# | | | |
| TST231 | 033644 | 11997# | | | |
| TST232 | 033714 | 11998# | | | |
| TST233 | 033770 | 11999# | | | |
| TST234 | 034040 | 12000# | | | |
| TST235 | 034110 | 12001# | | | |
| TST236 | 034160 | 12002# | | | |
| TST237 | 034234 | 12003# | | | |
| TST240 | 034304 | 12004# | | | |
| TST241 | 034354 | 12005# | | | |
| TST242 | 034444 | 12010# | | | |
| TST243 | 034514 | 12011# | | | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| TST244 | 034564 | 12012# | | | | | | |
| TST245 | 034634 | 12013# | | | | | | |
| TST246 | 034702 | 12014# | | | | | | |
| TST247 | 034750 | 12015# | | | | | | |
| TST250 | 035016 | 12016# | | | | | | |
| TST251 | 035066 | 12017# | | | | | | |
| TST252 | 035136 | 12018# | | | | | | |
| TST253 | 035204 | 12019# | | | | | | |
| TST254 | 035252 | 12030# | | | | | | |
| TST255 | 035326 | 12031# | | | | | | |
| TST256 | 035402 | 12032# | | | | | | |
| TST257 | 035462 | 12033# | | | | | | |
| TST260 | 035536 | 12034# | | | | | | |
| TST261 | 035612 | 12035# | | | | | | |
| TST262 | 035672 | 12036# | | | | | | |
| TST263 | 035746 | 12037# | | | | | | |
| TST264 | 036022 | 12038# | | | | | | |
| TST265 | 036076 | 12039# | | | | | | |
| TST266 | 036156 | 12040# | | | | | | |
| TST267 | 036232 | 12041# | | | | | | |
| TST270 | 036306 | 12042# | | | | | | |
| TST271 | 036366 | 12043# | | | | | | |
| TST272 | 036442 | 12044# | | | | | | |
| TST273 | 036516 | 12045# | | | | | | |
| TST274 | 036564 | 12046# | | | | | | |
| TST275 | 036642 | 12051# | | | | | | |
| TST276 | 036716 | 12052# | | | | | | |
| TST277 | 036772 | 12053# | | | | | | |
| TST300 | 037046 | 12054# | | | | | | |
| TST301 | 037120 | 12055# | | | | | | |
| TST302 | 037172 | 12056# | | | | | | |
| TST303 | 037244 | 12057# | | | | | | |
| TST304 | 037320 | 12058# | | | | | | |
| TST305 | 037374 | 12059# | | | | | | |
| TST306 | 037446 | 12060# | | | | | | |
| TST37 | 027006 | 11383 | 11414 | 11445 | 11476 | 11509 | 11542 | 11548# |
| TST40 | 027042 | 11549 | 11555# | | | | | |
| TST41 | 027056 | 11556 | 11559# | | | | | |
| TST42 | 027074 | 11560 | 11563# | | | | | |
| TST43 | 027130 | 11564 | 11570# | | | | | |
| TST44 | 027162 | 11571 | 11576# | | | | | |
| TST45 | 027214 | 11577 | 11582# | | | | | |
| TST46 | 027246 | 11583 | 11588# | | | | | |
| TST47 | 027304 | 11589 | 11595# | | | | | |
| TST50 | 027334 | 11596 | 11601# | | | | | |
| TST51 | 027406 | 11613# | | | | | | |
| TST52 | 027446 | 11614# | | | | | | |
| TST53 | 027506 | 11615# | | | | | | |
| TST54 | 027546 | 11616# | | | | | | |
| TST55 | 027604 | 11617# | | | | | | |
| TST56 | 027642 | 11618# | | | | | | |
| TST57 | 027700 | 11619# | | | | | | |
| TST60 | 027740 | 11620# | | | | | | |
| TST61 | 030000 | 11621# | | | | | | |
| TST62 | 030036 | 11622# | | | | | | |
| TS1 | 001616 | 6517 | 65510 | | | | | |

| | | | |
|---|---|---|---|
| TS10 | 003116 | 6663# | |
| TS100 | 006166 | 7563 | 7573# |
| TS101 | 006210 | 7581 | 7591# |
| TS102 | 006234 | 7600 | 7609# |
| TS103 | 006304 | 7627 | 7636# |
| TS104 | 006340 | 7648 | 7658# |
| TS105 | 006424 | 7680 | 7689# |
| TS106 | 006464 | 7703 | 7714# |
| TS107 | 006546 | 7736 | 7745# |
| TS11 | 003312 | 6685# | |
| TS110 | 006600 | 7756 | 7765# |
| TS111 | 006640 | 7779 | 7788# |
| TS112 | 006676 | 7800 | 7810# |
| TS113 | 006734 | 7822 | 7830# |
| TS114 | 006752 | 7836 | 7842# |
| TS115 | 006770 | 7848 | 7854# |
| TS116 | 007020 | 7864 | 7874# |
| TS117 | 007114 | 7899 | 7906# |
| TS12 | 003330 | 6689 | 6691# |
| TS120 | 007160 | 7919 | 7926# |
| TS121 | 007236 | 7944 | 7950# |
| TS122 | 007276 | 7963 | 7971# |
| TS123 | 007316 | 7978 | 7986# |
| TS124 | 007336 | 7993 | 8002# |
| TS125 | 007356 | 8009 | 8022# |
| TS126 | 007412 | 8034 | 8043# |
| TS127 | 007434 | 8051 | 8060# |
| TS13 | 003350 | 6695 | 6697# |
| TS130 | 007462 | 8069 | 8080# |
| TS131 | 007512 | 8088 | 8096# |
| TS132 | 007544 | 8107 | 8116# |
| TS133 | 007570 | 8121 | 8129# |
| TS134 | 007614 | 8134 | 8142# |
| TS135 | 007640 | 8147 | 8149# |
| TS136 | 007672 | 8159 | 8161# |
| TS137 | 007724 | 8171 | 8173# |
| TS14 | 003370 | 6701 | 6703# |
| TS140 | 007770 | 8185 | 8187# |
| TS141 | 010064 | 8208 | 8211# |
| TS142 | 010164 | 8234 | 8236# |
| TS143 | 010240 | 8250 | 8252# |
| TS144 | 010334 | 8271 | 8273# |
| TS145 | 010412 | 8287 | 8289# |
| TS146 | 010466 | 8302 | 8304# |
| TS147 | 010544 | 8317 | 8325# |
| TS15 | 003410 | 6706 | 6722# |
| TS150 | 010574 | 8335 | 8343# |
| TS151 | 010636 | 8356 | 8363# |
| TS152 | 010724 | 8379 | 8386# |
| TS153 | 011034 | 8403 | 8412# |
| TS154 | 011076 | 8423 | 8433# |
| TS155 | 011164 | 8449 | 8460# |
| TS156 | 011260 | 8477 | 8487# |
| TS157 | 011362 | 8504 | 8513# |
| TS16 | 003436 | 6727 | 6729# |
| TS160 | 011462 | 8529 | 8543# |

| | | | |
|---|---|---|---|
| TS161 | 011530 | 8550 | 8567# |
| TS162 | 011576 | 8574 | 8590# |
| TS163 | 011644 | 8597 | 8608# |
| TS164 | 011712 | 8615 | 8624# |
| TS165 | 011762 | 8638 | 8649# |
| TS166 | 012072 | 8673 | 8682# |
| TS167 | 012216 | 8712 | 8721# |
| TS17 | 003470 | 6735 | 6737# |
| TS170 | 012326 | 8739 | 8749# |
| TS171 | 012364 | 8758 | 8770# |
| TS172 | 012432 | 8780 | 8790# |
| TS173 | 012462 | 8796 | 8806# |
| TS174 | 012522 | 8813 | 8824# |
| TS175 | 012544 | 8829 | 8838# |
| TS176 | 012570 | 8843 | 8853# |
| TS177 | 012624 | 8860 | 8870# |
| TS2 | 002040 | 6626# | |
| TS20 | 003522 | 6745 | 6747# |
| TS200 | 012650 | 8874 | 8884# |
| TS201 | 012704 | 8891 | 8903# |
| TS202 | 012746 | 8910 | 8924# |
| TS203 | 013000 | 8929 | 8943# |
| TS204 | 013034 | 8948 | 8984# |
| TS205 | 013302 | 9031 | 9050# |
| TS206 | 013670 | 9131 | 9141# |
| TS207 | 013720 | 9150 | 9165# |
| TS21 | 003550 | 6752 | 6754# |
| TS210 | 013760 | 9178 | 9179# |
| TS211 | 014024 | 9193 | 9194# |
| TS212 | 014070 | 9207 | 9208# |
| TS213 | 014136 | 9223 | 9236# |
| TS214 | 014222 | 9258 | 9260# |
| TS215 | 014324 | 9287 | 9299# |
| TS216 | 014344 | 9306 | 9308# |
| TS217 | 014404 | 9322 | 9323# |
| TS22 | 003602 | 6762 | 6784# |
| TS220 | 014446 | 9337 | 9349# |
| TS221 | 014564 | 9378 | 9380# |
| TS222 | 014650 | 9401 | 9413# |
| TS223 | 014742 | 9437 | 9439# |
| TS224 | 015062 | 9467 | 9469# |
| TS225 | 015104 | 9476 | 9489# |
| TS226 | 015216 | 9517 | 9519# |
| TS227 | 015324 | 9548 | 9560# |
| TS23 | 003614 | 6788 | 6790# |
| TS230 | 015432 | 9589 | 9590# |
| TS231 | 015534 | 9618 | 9619# |
| TS232 | 015644 | 9649 | 9650# |
| TS233 | 015764 | 9683 | 9686# |
| TS234 | 016132 | 9723 | 9726# |
| TS235 | 016200 | 9740 | 9743# |
| TS236 | 016300 | 9766 | 9776# |
| TS237 | 016362 | 9798 | 9808# |
| TS24 | 003630 | 6793 | 6795# |
| TS240 | 016450 | 9829 | 9837# |
| TS241 | 016512 | 9851 | 9859# |

```
TS242   016634          9885   9896#
TS243   016700          9909   9911#
TS244   016736          9919   9921#
TS245   017002          9930   9932#
TS246   017042          9940   9942#
TS247   017100          9950   9952#
TS25    003644          6798   6800#
TS250   017144          9960   9962#
TS251   017210          9970   9980#
TS252   017256          9991   9993#
TS253   017326         10005  10007#
TS254   017402         10019  10021#
TS255   017456         10033  10035#
TS256   017532         10047  10049#
TS257   017610         10061  10063#
TS26    003660          6803   6805#
TS260   017666         10075  10085#
TS261   017726         10086  10090  10098#
TS262   017766         10117#
TS263   020100         10142#
TS264   020160         10157#
TS265   020242         10170  10181#
TS266   020374         10218#
TS267   020432         10234#
TS27    003706          6813   6815#
TS270   020454         10241  10256#
TS271   020646         10299#
TS272   020702         10310  10312#
TS273   020736         10323  10325#
TS274   021026         10343  10345#
TS275   021116         10364  10374#
TS276   021134         10378  10380#
TS277   021156         10384  10386#
TS3     002162          6627#
TS30    003740          6824   6825#
TS300   021206         10391  10393#
TS301   021232         10397  10399#
TS302   021254         10403  10409#
TS303   021272         10414  10420#
TS304   021322         10426  10428#
TS305   021364         10436  10438#
TS306   021616         10523#
TS307   022016         10571  10572#
TS31    003766          6833   6835#
TS310   022256         10611  10612#
TS311   022376         10634  10637#
TS312   022414         10644#
TS313   022440         10649  10651#
TS314   022530         10678#
TS315   022564         10679#
TS316   022620         10681#
TS317   022654         10689#
TS32    004020          6844   6846#
TS320   022710         10690#
TS321   022744         10691#
TS322   023010         10693  10694#
```

```
TS323   023044          10696#
TS324   023126          10712#
TS325   023156          10721#
TS326   023216          10730   10731#
TS327   023254          10739   10742#
TS33    004046           6854    6856#
TS330   023310          10751   10754#
TS331   023364          10771#
TS332   023424          10782   10784#
TS333   023510          10800#
TS334   023560          10822#
TS335   024020          10878#
TS336   024120          10893#
TS337   024230          10909#
TS34    004100           6865    6867#
TS340   024362          10929#
TS341   024426          10930   10935   10937#
TS342   024514          10938   10952#
TS343   024704          10982#
TS344   024736          10992   10997#
TS345   024766          11003   11010#
TS346   025016          11016   11026#
TS347   025134          11052#
TS35    004126           6875    6877#
TS350   025176          11063#
TS351   037670          12298#
TS352   037724          12317#
TS353   037766          12338#
TS354   040054          12356   12369#
TS355   040112          12376   12404#
TS356   040136          12416#
TS357   040156          12422#
TS36    004160           6886    6889#
TS360   040176          12428#
TS361   040216          12434#
TS362   040236          12447#
TS363   040354          12495#
TS364   040410          12515#
TS365   040472          12531   12544#
TS366   040562          12569   12581#
TS367   040660          12608   12621#
TS37    004206           6897    6899#
TS370   040764          12649   12660#
TS371   041070          12687   12702#
TS372   041234          12743#
TS373   041420          12746   12832#
TS374   042074          12886   12905#
TS375   042424          12937#
TS376   042772          12999   13010#
TS377   043146          13016#
TS4     002402           6645#
TS40    004240           6908    6910#
TS400   043334          13030#
TS401   043456          13079#
TS402   043734          13098   13145#
TS403   044124          13232#
```

| | | | | |
|---|---|---|---|---|
| TS404 | 044424 | 13255 | 13285 | 13333# |
| TS405 | 044710 | 13354 | 13384 | 13435# |
| TS406 | 045064 | 13481# | | |
| TS407 | 045440 | 13562# | | |
| TS41 | 004266 | 6918 | 6920# | |
| TS410 | 045576 | 13596# | | |
| TS411 | 045642 | 13607 | 13624# | |
| TS412 | 045774 | 13670# | | |
| TS413 | 046300 | 13744# | | |
| TS414 | 046774 | 13881# | | |
| TS415 | 047526 | 13939 | 14021# | |
| TS416 | 050126 | 14135# | | |
| TS417 | 050210 | 14168# | | |
| TS42 | 004320 | 6929 | 6945# | |
| TS420 | 050756 | 14364# | | |
| TS421 | 051174 | 14405# | | |
| TS422 | 051322 | 14432# | | |
| TS423 | 051774 | 14522# | | |
| TS424 | 052272 | 14574# | | |
| TS425 | 052704 | 14664# | | |
| TS426 | 053012 | 14691# | | |
| TS427 | 053156 | 14729# | | |
| TS43 | 004342 | 6949 | 6951# | |
| TS430 | 053514 | 14800# | | |
| TS431 | 054114 | 14881# | | |
| TS432 | 054654 | 15108# | | |
| TS433 | 054746 | 15139# | | |
| TS434 | 055002 | 15165# | | |
| TS435 | 055232 | 15256# | | |
| TS436 | 055400 | 15309# | | |
| TS437 | 055446 | 15339# | | |
| TS44 | 004362 | 6954 | 6956# | |
| TS440 | 056006 | 15440# | | |
| TS441 | 056310 | 15538# | | |
| TS442 | 056614 | 15645# | | |
| TS443 | 060424 | 15739# | | |
| TS444 | 061312 | 15830# | | |
| TS445 | 061620 | 15924# | | |
| TS446 | 061764 | 15984# | | |
| TS447 | 062130 | 16041# | | |
| TS45 | 004402 | 6959 | 6961# | |
| TS450 | 062326 | 16111# | | |
| TS451 | 062464 | 16166# | | |
| TS452 | 062640 | 16226# | | |
| TS453 | 063014 | 16287# | | |
| TS454 | 063136 | 16335# | | |
| TS455 | 063270 | 16398# | | |
| TS456 | 064056 | 16540# | | |
| TS457 | 064432 | 16634# | | |
| TS46 | 004422 | 6964 | 6996# | |
| TS460 | 065026 | 16735# | | |
| TS461 | 065256 | 16800# | | |
| TS462 | 065470 | 16863# | | |
| TS463 | 065774 | 16962# | | |
| TS464 | 066604 | 17128# | | |
| TS465 | 067404 | 17287# | | |

L 7



```
TS466   070304          17464#
TS467   070542          17536#
TS47    004450           7004   7016#
TS470   070766          17622#
TS471   071542          17790#
TS472   072544          17993#
TS473   073406          18183#
TS474   074212          18417#
TS475   074466          18504#
TS476   075220          18678#
TS477   075712          18812#
TS5     002524           6646#
TS50    004506           7029   7037#
TS500   076444          18985#
TS501   076776          19092#
TS502   077300          19221#
TS503   077662          19357#
TS504   100214          19489#
TS505   100626          19630#
TS506   101562          19846#
TS507   102776          20067#
TS51    004526           7043   7053#
TS510   103436          20217#
TS511   104142          20357#
TS512   105040          20493#
TS513   105134          20511  20534#
TS514   105316          20588#
TS515   105604          20663#
TS516   105756          20720#
TS517   106142          20776#
TS52    004552           7061   7070#
TS520   106306          20825#
TS521   106452          20871#
TS522   106610          20913#
TS523   106764          20960#
TS524   107416          21135#
TS525   110050          21305#
TS526   110110          21328#
TS527   110230          21371#
TS53    004606           7082   7094#
TS530   110270          21395#
TS531   110330          21420#
TS532   110464          21467#
TS533   110622          21509#
TS534   110750          21547#
TS535   111106          21590#
TS536   111254          21631#
TS537   111422          21673#
TS54    004646           7108   7120#
TS540   111554          21710#
TS541   111724          21751#
TS542   112042          21785#
TS543   112200          21830#
TS544   112324          21869#
TS545   112454          21907#
TS546   112604          21946#
```

```
TS547   112746          21986#
TS55    004710           7135   7147#
TS550   113106          22028#
TS551   113250          22068#
TS552   113400          22106#
TS553   113534          22142#
TS554   113676          22185#
TS555   114050          22234#
TS556   114770          22498#
TS557   115052          22525#
TS56    004752           7162   7169#
TS560   115126          22554#
TS561   115226          22574#
TS562   115334          22596#
TS563   115434          22618#
TS564   115530          22640#
TS565   115640          22672#
TS566   115704          22697#
TS567   116002          22730#
TS57    005022           7187   7194#
TS570   117002          22969#
TS571   117556          23121#
TS572   121062          23372#
TS573   121174          23401#
TS574   121306          23431#
TS575   121420          23459#
TS576   121542          23489#
TS577   121666          23519#
TS6     002652           6655#
TS60    005074           7213   7214#
TS600   122012          23551#
TS601   122150          23588#
TS602   122226          23612#
TS603   122304          23640#
TS604   123404          23880#
TS605   123444          23901#
TS606   124002          24024#
TS607   124110          24055#
TS61    005154           7233   7234#
TS610   124200          24076#
TS611   124670          24196#
TS612   124716          24205#
TS613   124744          24213#
TS614   125032          24235#
TS615   125064          24243   24247#
TS616   125112          24255#
TS617   125140          24265#
TS62    005226           7251   7267#
TS620   125256          24286#
TS621   125350          24305#
TS622   125450          24323#
TS623   125566          24346#
TS624   126020          24388#
TS625   126046          24395#
TS626   126202          24419#
TS627   126372          24459#
```

```
TS63    005262          7279    7293#
TS630   126522         24460   24486#
TS631   126772         24487   24537#
TS632   127020         24546#
TS633   127046         24554#
TS634   127130         24555   24572#
TS635   127200         24573   24581   24585#
TS636   127226         24593#
TS637   127254         24605#
TS64    005332          7311    7327#
TS640   127366         24614   24618   24621#
TS641   127522         24645#
TS642   127656         24670#
TS643   127750         24689#
TS644   130050         24707#
TS645   130166         24730#
TS646   130312         24731   24752#
TS647   130376         24774#
TS65    005376          7344    7345#
TS650   130424         24780   24784#
TS651   130534         24804#
TS652   130636         24823#
TS653   130762         24847#
TS654   131050         24864#
TS655   131162         24865   24882#
TS656   131242         24899#
TS657   131336         24900   24918#
TS66    005500          7369    7379#
TS660   131414         24930#
TS661   131650         24981#
TS662   131776         24982   25013#
TS67    005536          7392    7409#
TS7     002774          6662#
TS70    005604          7425    7435#
TS71    005640          7444    7455#
TS72    005676          7465    7467#
TS73    005734          7479    7480#
TS74    006004          7497    7498#
TS75    006044          7511    7512#
TS76    006110          7526    7536#
TS77    006144          7544    7555#
TTBDON  113374         22092   22105#
TTBTP1  113352         22069   22074   22082   22088   22095#
TTBTP2  113362         22070   22083   22099#
TTB10   113372         22086   22089   22091   22104#
TTB2    113306         22079#
TTCDON  122144         23573   23580#
TTCSR = 177564          6383#  10872   10884*  10889*  10898*  10908*  10912*  10927*  10961*  10966   10969   10972*  10981*
                       11064*  24088   24091*  24100*  24102   24105*  24123*  25164   25168
TTCTB0  122122         23553   23576#
TTCTB1  122126         23562   23569   23571   23577#
TTCTB2  122136         23561   23567   23578#
TTC10   122142         23558   23566   23568   23570   23572   23579#
TTC2    122060         23564#
TTTA1   106264         20809#
TTTA2   106266         20783*  20789   20793   20810#
```

```
TTTBF0  106252          20778   20783   20795   20804#  20810
TTTDON  106302          20801   20818#
TTTTP1  106270          20786   20796   20811#
TTT10   106300          20788   20794   20799   20816#
TTT2    106216          20791#
TTT3    106242          20798#  20800
TVECT   124624          24180#  24288   24289#  24294#  24302#  24308   24309#  24316#  24320#  24325   24327#  24338#  24343#
                        24349   24350#  24360#  25023   25088#
TVECT2  126734          24526#  24622   24623#  24642#  24672   24673#  24678#  24686#  24692   24693#  24700#  24704#  24709
                        24711#  24722#  24727#  24734   24735#  24745#  25021   25029#  25086#
TYPE    133364           6499    6639   14346   14495   14784   14989   15001   24136   24373   24513   24958   25103   25121
                        25156   25160#
T105    056410          15567   15568#
T12     056412          15572#
T13     056426          15575#  15576
T14     056454          15586#
T15     056456          15587#
T16     056460          15588#
T17     056476          15593   15594#
T2      056326          15545#  15546
T20     056536          15605#  15607
T23     056544          15606   15607#
T3      056352          15556#
T4      056354          15557#
T5      056356          15558#
T6      056402          15566#  15568
UDONE   064056          16498   16530#
UERR0   063412          16407   16422#
UERR3   063744          16480   16492#
UFDFLG  001274           6484#  25172
UFDSET= 000001           6368#   6484
UFLAG   064042          16399#  16468   16475#  16524#
UIPAR0= 177640          12081#  12429   12570   12650   12727   12781   12944   13566#  13583#  13761#  14252   14305
UIPAR1= 177642          12081#  13762#
UIPAR2= 177644          12081#  13763#
UIPAR3= 177646          12081#  13018   13084#  13672#  13764#
UIPAR4= 177650          12081#  12961#  12981#  12996#  13085#  13673#  13769#  13885#  14031#
UIPAR5= 177652          12081#  12962#  12982#  12997#
UIPAR6= 177654          12081#
UIPAR7= 177656          12081#  12568   12648   13765#
UIPDR0= 177600          12081#  12435   12609   12688   12720   12770   12951   13018   13758   14025   14248   14293
UIPDR1= 177602          12081#
UIPDR2= 177604          12081#
UIPDR3= 177606          12081#  13087#  13674#  13723#
UIPDR4= 177610          12081#  12959#  12983#  13094#  13154#  13675#  13883#  14038#
UIPDR5= 177612          12081#  12960#  12984#
UIPDR6= 177614          12081#
UIPDR7= 177616          12081#  12607   12686
UNIQUE  131604          24532   24944   24972#
UPAT00  063772          16400   16410   16426   16442   16456   16471   16483   16499#
UPAT01  063774          16500#
UPAT02  063776          16501#
UPAT03  064000          16502#
UPAT10  064002          16416   16504#
UPAT11  064004          16505#
UPAT12  064006          16506#
```

```
UPAT13  064010          16507#
UPAT20  064012          16432   16509#
UPAT21  064014          16510#
UPAT22  064016          16511#
UPAT23  064020          16512#
UPAT30  064022          16447   16514#
UPAT31  064024          16515#
UPAT32  064026          16516#
UPAT33  064030          16517#
UPAT40  064032          16461   16485   16519#
UPAT41  064034          16520#
UPAT42  064036          16521#
UPAT43  064040          16522#
UQUIET  001276           6484#* 25172
UROM1   064050          16412*  16428*  16444*  16458*  16527#
UROM2   064052          16413*  16429*  16445*  16459*  16528#
UROM3   064054          16414*  16430*  16446*  16460*  16529#
USESTK= 000600           6401#  10145   10152   10161   10432   10435   10440   12373   13570   13585   13692   13782   13900
                        14041   14142   14157
USP1    017742          10102#  10103
USP1A   017760          10104   10107#
USP2    020016          10121   10124#
USP3    020034          10126   10127#
USP4    020062          10131   10133#
USRM  = 140000           6366#  10099   10105   10107   10118   10122   10124   10128   10130   10134   10144   10149   10151
                        10160   10164   10166
UTMP1   064044          16405*  16412   16458   16476*  16525#
UTMP2   064046          16406*  16428   16444   16477*  16526#
UUBDON  113530          22132   22141#
UUBTP1  113444          22108   22112   22113*  22119#  22122
UUBTP2  113516          22107   22123   22135#
UUB10   113526          22126   22129   22131   22140#
UUB2    113442          22118#
UUCBF0  122214          23593   23597   23602#
UUCDON  122222          23598   23604#
UUCTP1  122202          23591   23600#
UUC2    122170          23595#
UUUA1   106430          20832*  20856#
UUUA2   106432          20838   20840   20857#
UUUBF0  106416          20827   20832   20842   20851#  20856
UUUDON  106446          20848   20864#
UUUTP1  106434          20835   20843   20858#
UUU10   106444          20837   20841   20846   20863#
UUU2    106362          20839#
UUU3    106406          20845#  20847
U0      063304          16402#  16403
U1      063326          16407#  16478
U10     063616          16462#
U11     063620          16463#
U12     063636          16467   16468#
U13     063654          16473#  16474
U14     063706          16469   16480#
U15     063734          16486#
U16     063736          16487#  16492
U2      063376          16417#
U3      063414          16422   16423#
```

```
U4      063456          16434#
U5      063474          16439   16440#
U6      063536          16449#
U7      063554          16453   16454#
VDEC    022556          10678#
VDEC10  022732          10690#
VDEC11  022766          10691#
VDEC12  022772          10691#
VDEC13  023032          10694#
VDEC14  023036          10694#
VDEC2   022552          10678#
VDEC3   022612          10679#
VDEC4   022606          10679#
VDEC5   022646          10684   10687#
VDEC6   022642          10683   10686#
VDEC7   022702          10689#
VDEC8   022676          10689#
VDEC9   022736          10690#
VECT1   051744          14507#  14544
VMKOR   001300           6484#
VRTPCR  054114          14898#  14936*  14937   14960*  14961
VVBDON  110324          21408   21411#
VVB10   110322          21405   21407   21409#
VVB2    110276          21399#
VVCBF0  122272          23618   23622   23627#
VVCDON  122300          23623   23629#
VVCTP1  122260          23616   23625#
VVC2    122246          23620#
VVVBF0  106562          20875   20883   20887   20889   20896#
VVVDON  106604          20895   20906#
VVVTP1  106572          20881   20890   20900#
VVV10   106602          20880   20888   20893   20905#
VVV2    106524          20885#
WAITER  132332          25070#
WAITI0  132204          25041#  25042
WASR6   037556          12272#
WASSR0  037564          12272#  13108*  13110   13168*  13170   13260*  13263   13359*  13362   13508*  13510   13524*  13525
                        13636*  13639   13686*  13705
WASSR2  037566          12272#  12457*  12459   12475*  12476   13109*  13116   13169*  13176   13261*  13269   13360*  13368
                        13449*  13450   13485*  13487   13492*  13494   13509*  13513   13528*  13529   13538*  13540   13637*
                        13645   13687*  13708
WATE    024646          10964   10976#
WATE1   024600          10966#  10967
WATE2   024614          10969#  10970
WATE3   024642          10974#  10978
WBIT =  000100          12086#  13011   13018
WC      051762          14514#  14552*  14553*  14675*
WDAP00  064416          16620#
WDAT01  064420          16621#
WDAT02  064422          16622#
WDAT03  064424          16623#
WDONE   064426          16613   16625#
WDONE2  130320          24755#  24758
WPAT00  064406          16543   16545   16550   16552   16561   16565   16569   16571   16580   16582   16586   16588   16597
                        16601   16605   16607   16615#
WPAT01  064410          16616#
```

```
WPAT02  064412          16617#
WPAT03  064414          16618#
WWBDON  114764          22406   22488#
WWB1    114050          22236#
WWB10   114462          22369#
WWB11   114530          22388#
WWB2    114116          22255#
WWB3    114164          22274#
WWB4    114232          22293#
WWB5    114300          22312#
WWB6    114346          22331#
WWB7    114414          22350#
WWCDON  123400          23799   23870#
WWC1    122304          23643#
WWC10   122610          23707#
WWC11   122644          23717#
WWC12   122700          23726#
WWC13   122734          23735#
WWC14   122770          23744#
WWC15   123024          23753#
WWC16   123060          23762#
WWC17   123114          23771#
WWC2    122340          23652#
WWC20   123150          23781#
WWC21   123204          23791#
WWC3    122374          23661#
WWC4    122430          23670#
WWC5    122464          23680#
WWC6    122520          23689#
WWC7    122554          23698#
WWMBF0  106726          20917   20927   20932   20939#
WWMBF1  106746          20925   20927*  20930   20947#
WWMDON  106760          20938   20953#
WWMTP1  106736          20923   20933   20943#
WWM10   106756          20922   20931   20936   20952#
WWM2    106670          20928#
W10     064220          16574   16575#
W11     064232          16577   16578#
W12     064252          16583#
W13     064276          16590#  16592
W14     064304          16591   16592#
W15     064316          16594   16595#
W16     064342          16602#
W17     064366          16609#  16611
W2      064076          16546#
W20     064374          16610   16611#
W3      064122          16554#  16556
W4      064130          16555   16556#
W5      064142          16558   16559#
W6      064166          16566#
W7      064212          16573#  16575
X       124412          24119#
XAPT11  065004          16718#
XBUF    104724          20393   20395   20405   20407   20418   20420   20460   20462   20473#
XDAT00  064762          16643   16661   16679   16697   16707#
XDAT01  064764          16708#
```

```
XDAT02  064766          16709#
XDAT03  064770          16710#
XDONE   065022          16706   16726#
XMIT2   132220          25029   25047#
XMTCT2  132452          25026*  25047*  25050   25077   25081   25093#
XOR1    016416           9814    9815    9816    9817    9819#
XOR2    016420           9819    9820#
XOR3    016446           9824    9825    9826    9827    9829#
XPAT0   105024          20425*  20426   20431*  20432   20481#
XPAT0   104734          20368   20370   20376   20378   20387   20412   20437   20474#
XPAT00  064772          16637   16645   16673   16681   16712#
XPAT01  064774          16713#
XPAT02  064776          16714#
XPAT03  065000          16715#
XPAT1   104744          20475#
XPAT10  065002          16639   16657   16675   16693   16717#
XPAT12  065006          16719#
XPAT13  065010          16720#
XPAT2   104754          20459   20476#
XPAT20  065012          16655   16663   16691   16699   16721#
XPAT21  065014          16722#
XPAT22  065016          16723#
XPAT23  065020          16724#
XPAT3   104764          20388   20400   20413   20477#
XPAT4   104774          20424   20430   20438   20444   20445   20451   20452   20458   20478#
XPAT5   105004          20466   20479#
XPAT6   105014          20467   20480#
XTDONE  105034          20470   20483#
XT1     104142          20358#
XT1A    104156          20358   20361#
XT10    104566          20441   20443#
XT11    104614          20448   20450#
XT12    104642          20455   20457#
XT13    104676          20463   20465#
XT2     104160          20360   20363#
XT2A    104200          20367   20368#
XT2B    104216          20371   20372#
XT3     104230          20374   20375#
XT3A    104252          20379   20380#
XT4     104262          20382   20384#
XT4A    104312          20385   20390#
XT4B    104324          20392   20393#
XT5     104342          20396   20398#
XT5A    104362          20398   20402#
XT5B    104374          20404   20405#
XT6     104412          20408   20410#
XT6A    104436          20410   20415#
XT6B    104450          20417   20418#
XT7     104466          20421   20423#
XT8     104514          20427   20429#
XT9     104542          20433   20436#
XXBDON  113672          22169   22183#
XXBTP1  113650          22143   22148   22149*  22159   22165   22172#
XXBTP2  113660          22144   22160   22176#
XXB10   113670          22158   22163   22166   22168   22182#
XXB2    113576          22154#
```

```
XXCDON  123440          23892#
XXXDON  107412          21056   21128#
XXX1    106764          20963#
XXX2    107030          20982#
XXX3    107074          21001#
XXX4    107140          21020#
XXX5    107204          21039#
X10     064572          16666   16667#
X11     064606          16670   16671#
X12     064626          16676#
X13     064652          16683#  16685
X14     064660          16684   16685#
X15     064674          16688   16689#
X16     064714          16694#
X17     064740          16701#  16703
X2      064452          16640#
X20     064746          16702   16703#
X3      064476          16647#  16649
X4      064504          16648   16649#
X5      064520          16652   16653#
X6      064540          16658#
X7      064564          16665#  16667
Y       124414          24096   24120#
YBR     001760           6563*   6566*  65760
YDAT00  065212          16748   16771#
YDAT01  065214          16772#
YDAT02  065216          16773#
YDAT03  065220          16774#
YDONE   065252          16764   16791#
YFLAG   065202          16736*  16757   16759*  16766#
YNTAB   021470           6553   10466#
YPAT00  065222          16737   16761   16776#
YPAT01  065224          16777#
YPAT02  065226          16778#
YPAT03  065230          16779#
YPAT10  065232          16738   16760   16781#
YPAT11  065234          16782#
YPAT12  065236          16783#
YPAT13  065240          16784#
YPAT20  065242          16742   16786#
YPAT21  065244          16787#
YPAT22  065246          16788#
YPAT23  065250          16789#
YTMP1   065204          16737*  16744   16760*  16767#
YTMP2   065206          16738*  16751   16761*  16768#
YTMP3   065210          16739*  16755   16762*  16769#
YYBDON  114044          22214   22228#
YYBTP1  114020          22187   22193   22194*  22204   22217#  22225
YYBTP2  114030          22188   22205   22221#
YYBTP3  114040          22192   22210   22225#
YYB10   114042          22203   22208   22211   22213   22226#
YYB2    113746          22199#
YYCDON  123776          23954   24012#
YYC1    123444          23904#
YYC2    123472          23912#
YYC3    123520          23920#
```

```
YYC4    123546          23929#
YYC5    123574          23938#
YYC6    123622          23947#
YYYDON  110044          21231   21299#
YYY1    107416          21138#
YYY2    107462          21157#
YYY3    107526          21176#
YYY4    107572          21195#
YYY5    107636          21214#
Y1      065054          16740#  16763
Y2      065074          16745#
Y3      065120          16752#  16754
Y4      065140          16756   16757#
Y5      065200          16758   16764#
Z       124416          24097*  24115*  24116   24121#
ZDAT00  065424          16812   16833#
ZDAT01  065426          16834#
ZDAT02  065430          16835#
ZDAT03  065432          16836#
ZDONE   065464          16827   16853#
ZFLAG   065416          16801*  16821   16823*  16829#
ZPAT00  065434          16802   16838#
ZPAT01  065436          16839#
ZPAT02  065440          16840#
ZPAT03  065442          16841#
ZPAT10  065444          16824   16843#
ZPAT11  065446          16844#
ZPAT12  065450          16845#
ZPAT13  065452          16846#
ZPAT20  065454          16806   16848#
ZPAT21  065456          16849#
ZPAT22  065460          16850#
ZPAT23  065462          16851#
ZTMP1   065420          16802*  16808   16815   16824*  16830#
ZTMP2   065422          16803*  16819   16825*  16831#
ZZCBF   124072          24031   24037   24049#
ZZCDON  124104          24044   24052#
ZZC10   124066          24039   24041   24043   24045#
ZZC2    124010          24029#  24040
ZZC3    124030          24034#
ZZZDON  110104          21317   21321#
ZZZ10   110102          21314   21316   21319#
ZZZ2    110056          21309#
Z1      065276          16804#  16826
Z2      065316          16809#
Z3      065342          16816#  16818
Z4      065350          16817   16818#
Z5      065362          16820   16821#
Z6      065414          16822   16827#
$APTHD  001030           6425#
$BDADR  131464          24946#
$BDDAT  131466          24947#
$CPUOP  001026           6424#
$DEVCT  001010           6424#
$DOAGN  133310          25116   25137#
$ENDAD  133226           6422   25125#  25172
```

```
$ENV    001020          6424#  6494   6516   6636   10086  10520  10930  10938  10956  12746  14327  14343  14488
                       14781  14986  14998  24133  24214  24370  24407  24460  24487  24510  24555  24573  24614
                       24633  24657  24731  24760  24865  24900  24955  24982  25014  25100
$ENVM   001021          6424#  25160
$EOPCT  133162         25115#  25119#
$ETABL  001020          6424#
$ETEND  001030          6424#  6425
$FATAL  001002          6424#  6511#  6634#  14341#  14486#  14779#  14984#  14996#  24131#  24368#  24508#  24953#  25098#
$GDADR  131470         24948#
$GDDAT  131472         24949#
$HIBTS  001030          6425#
$MAIL   001000          6424#  6425
$MBADR  001032          6425#
$MSGAD  001014          6424#
$MSGLG  001016          6424#
$MSGTY  001000          6424#  6512#  6635#  14342#  14487#  14780#  14985#  14997#  24132#  24369#  24509#  24954#  25099#
$PASS   001006          6424#  6500#  10086  10930  10938  10958  11352  11385  11416  11447  11479  11512  11651
                       11690  11729  12746  14329  24216  24409  24460  24487  24555  24573  24614  24635  24659
                       24731  24762  24865  24900  24982  25016  25117#  25118#
$PASTM  001036          6425#
$SETUP= 000020          6484#
$STUP = 177777          6484#
$SVPC = 000400          6422#
$SWR  = 000000          6351#
$SWREG  001022          6424#  10522
$TESTN  001004          6424#  6477   6510#  10210  11323  12287#  14373#  14536#  14806#  15095#  24192#  24384#  24533#
                       24978#
$TMP0   037572         12272#  12847#  12848#  12849#  12909#  12910#  12911#  12912#  12963#  13096#  13122  13156#  13182
                       13503#  13510  13694#  13700#  13704#  13707#  13711#  13712  24082#  24124  25021#  25086  25131#
                       25136
$TMP1   037574         12272#  13500#  13512#  13515#  13516  13521#  13527#  13531#  13532  24083#  24125  25022#  25087
$TMP2   037576         12272#  13504#  13513  13832#  13833  13855#  13856  13962  13963#  13993#  13995  14093#  14094
                       14116#  14117  20499#  20643#  22504#  22559#  25023#  25088
$TMP3   037600         12272#  25024#  25089
$TMP4   037602         12272#  25025#  25090
$TN   = 000663          6352#  6551#  6626#  6627#  6645#  6646#  6655#  6662#  6663#  6685#  6689  6691#  6695
                        6697#  6701  6703#  6706  6722#  6727  6729#  6735  6737#  6745  6747#  6752  6754#
                        6762  6784#  6788  6790#  6793  6795#  6798  6800#  6803  6805#  6813  6815#  6824
                        6825#  6833  6835#  6844  6846#  6854  6856#  6865  6867#  6875  6877#  6886  6889#
                        6897  6899#  6908  6910#  6918  6920#  6929  6945#  6949  6951#  6954  6956#  6959
                        6961#  6964  6996#  7004  7016#  7029  7037#  7043  7053#  7061  7070#  7082  7094#
                        7108  7120#  7135  7147#  7162  7169#  7187  7194#  7213  7214#  7233  7234#  7251
                        7267#  7279  7293#  7311  7327#  7344  7345#  7369  7379#  7392  7409#  7425  7435#
                        7444  7455#  7465  7467#  7479  7480#  7497  7498#  7511  7512#  7526  7536#  7544
                        7555#  7563  7573#  7581  7591#  7600  7609#  7627  7636#  7648  7658#  7680  7689#
                        7703  7714#  7736  7745#  7756  7765#  7779  7788#  7800  7810#  7822  7830#  7836
                        7842#  7848  7854#  7864  7874#  7899  7906#  7919  7926#  7944  7950#  7963  7971#
                        7978  7986#  7993  8002#  8009  8022#  8034  8043#  8051  8060#  8069  8080#  8088
                        8096#  8107  8116#  8121  8129#  8134  8142#  8147  8149#  8159  8161#  8171  8173#
                        8185  8187#  8208  8211#  8234  8236#  8250  8252#  8271  8273#  8287  8289#  8302
                        8304#  8317  8325#  8335  8343#  8356  8363#  8379  8386#  8403  8412#  8423  8433#
                        8449  8460#  8477  8487#  8504  8513#  8529  8543#  8550  8567#  8574  8590#  8597
                        8608#  8615  8624#  8638  8649#  8673  8682#  8712  8721#  8739  8749#  8758  8770#
                        8780  8790#  8796  8806#  8813  8824#  8829  8838#  8843  8853#  8860  8870#  8874
                        8884#  8891  8903#  8910  8924#  8929  8943#  8948  8984#  9031  9050#  9131  9141#
                        9150  9165#  9178  9179#  9193  9194#  9207  9208#  9223  9236#  9258  9260#  9287
```

```
$TPB    025672
$TPS    025674
$TSTM   001034
$UNIT   001012
$UNITM  001040
$USWR   001024
$X  =   131776

 25119
 2478#  24804  24823  24847  24864  24865  24882  24899  24900  24918  24930  24981  24982
 25013#
 11202#
 11203#
 64254
 64244
 64254
 64244
 64254
 65514  66266  66274  66454  66464  66554  66624  66734  66854  66914  67034  67224
 67294  67374  67474  67544  67904  67954  68104  68464  68554  68914  69704  68464
 68564  68674  68774  68894  68994  69104  69264  69454  69564  69614  69964  70164
 70374  70534  70704  70944  71204  71474  71694  71934  72144  72344  72674  73274
 73454  73794  74094  74354  74554  74674  77454  77144  77654  77954  78424  78744
 76094  76364  76584  76894  77144  77454  77884  78104  78304  78424  78544  79124
 79064  79264  79504  79714  79864  80024  80434  80804  80964  81164  81294  81744
 81424  81494  81614  81734  81874  82114  82524  82734  82894  83044  83254  83434
 83634  83864  84124  84334  84604  84874  85434  85674  85904  86084  86244  86494
 86824  87214  87494  87704  87904  88064  88384  88534  88704  88844  89034  89244
 89434  89844  90504  91414  91794  92084  92364  92994  93204  93244  93084  92244
 93494  93804  94134  94394  94694  94894  95194  95604  95904  96194  96504  96864
 97434  97764  98084  98374  98594  99214  99324  99424  99524  99624  99624  99804
 99934  10007#  10021#  10035#  10049#  10063#  10098#  10117#  10154#  10174#  10225#  10262#
 10234#  10256#  10298#  10312#  10325#  10374#  10386#  10426#  10454#  10474#  10525#  10542#
 10428#  10438#  10523#  10572#  10612#  10644#  10678#  10711#  10754#  10810#  10852#  10894#
 10691#  10694#  10696#  10712#  10721#  10742#  10771#  10826#  10851#  10897#  10941#  10981#
 10893#  10909#  10929#  10937#  10952#  10997#  11026#  11052#  11063#  11106#  11152#  11198#
 12369#  12404#  12416#  12422#  12428#  12434#  12447#  12495#  12515#  12544#  12581#  12621#
 12338#
```

```
                    12660#  12702#  12743#  12832#  12905#  12937#  13010#  13016#  13030#  13079#  13145#  13232#  13333#
                    13435#  13481#  13562#  13596#  13624#  13670#  13744#  13881#  14021#  14135#  14168#  14347#  14363#
                    14364#  14405#  14432#  14496#  14503#  14522#  14574#  14664#  14691#  14729#  14785#  14787#  14800#
                    14881#  14990#  15004#  15108#  15139#  15165#  15256#  15309#  15339#  15440#  15538#  15645#  15739#
                    15830#  15924#  15984#  16041#  16111#  16123#  16166#  16226#  16287#  16335#  16398#  16540#  16634#
                    16735#  16800#  16863#  16962#  17128#  17287#  17464#  17536#  17622#  17790#  17993#  18183#  18417#
                    18504#  18678#  18812#  18985#  19092#  19221#  19357#  19489#  19630#  19846#  20067#  20217#  20357#
                    20493#  20534#  20588#  20663#  20720#  20776#  20825#  20871#  20913#  20960#  21135#  21305#  21328#
                    21371#  21395#  21420#  21467#  21509#  21547#  21590#  21631#  21673#  21710#  21751#  21785#  21830#
                    21869#  21907#  21946#  21986#  22028#  22068#  22106#  22142#  22185#  22234#  22498#  22525#  22554#
                    22574#  22596#  22618#  22640#  22672#  22697#  22730#  22969#  23121#  23372#  23401#  23431#  23459#
                    23489#  23519#  23551#  23588#  23612#  23640#  23880#  23901#  24024#  24055#  24076#  24137#  24196#
                    24205#  24213#  24235#  24247#  24255#  24265#  24286#  24305#  24323#  24346#  24374#  24377#  24388#
                    24395#  24419#  24459#  24486#  24514#  24537#  24546#  24554#  24572#  24585#  24593#  24605#  24621#
                    24645#  24670#  24689#  24707#  24730#  24752#  24774#  24784#  24804#  24823#  24847#  24864#  24882#
                    24899#  24918#  24930#  24959#  24981#  25013#  25104#  25107#  25109#
.RSET   124552      15131   15153   15234   15300   15324   15432   15530   15631   15730   15820   15916   15975   16032
                    16101   16157   16217   16279   16326   16382   16625   16726   16791   16853   16951   17116   17279
```

```
.       = 133657    6406#   6410    6412#   6413    6415#   6419    6422#   6423#   6425#   6429#   6438#   6441#   6444#
                    6447#   6453#   6461#   6466#   6471#   6474#   6481#   6551    6561    6580    6582    6626    6627
                    6640    6645    6646    6655    6662    6663    6685    6691    6697    6703    6722    6729    6737
                    6747    6754    6784    6790    6795    6800    6805    6815    6825    6835    6846    6856    6867
                    6877    6889    6899    6910    6920    6945    6951    6956    6961    6996    7016    7037    7053
                    7070    7094    7120    7147    7169    7194    7214    7234    7267    7293    7327    7345    7379
                    7409    7435    7455    7467    7480    7498    7512    7536    7555    7573    7591    7609    7636
                    7658    7689    7714    7745    7765    7788    7810    7830    7842    7854    7874    7906    7926
                    7950    7971    7986    8002    8022    8043    8060    8080    8096    8116    8129    8142    8149
                    8161    8173    8187    8211    8236    8252    8273    8289    8304    8325    8343    8363    8386
                    8412    8433    8460    8487    8513    8543    8567    8590    8608    8624    8649    8682    8721
                    8749    8770    8790    8806    8824    8838    8853    8870    8884    8903    8924    8943    8984
                    8988    9050    9141    9165    9179    9194    9208    9236    9260    9299    9308    9323    9349
                    9380    9413    9439    9469    9489    9519    9560    9590    9619    9650    9686    9726    9743
                    9776    9808    9837    9859    9896    9911    9921    9932    9942    9952    9962    9980    9993
                    10007   10021   10035   10049   10063   10085   10098   10117   10142   10157   10181   10218   10234
                    10256   10299   10312   10325   10345   10374   10380   10386   10393   10399   10409   10420   10428
                    10438   10449   10450   10451   10452   10453   10454   10455   10456   10457   10458   10459   10460
                    10461   10462   10463   10523   10572   10612   10637   10644   10651   10678   10679   10681   10689
                    10690   10691   10694   10696   10712   10716   10721   10725   10731   10735   10738   10742   10754
                    10771   10784   10792   10800   10822   10878   10893   10909   10929   10937   10952   10982   10997
                    11010   11026   11052   11063   11164   11178#  11180#  11182#  11184#  11186#  11188#  12298   12317
                    12338   12369   12404   12416   12422   12428   12434   12447   12495   12515   12544   12581   12621
```

```
                    17456   17526   17601   17774   17985   18171   18405   18493   18668   18801   18974   19080   19208
                    19341   19477   19617   19834   20055   20205   20349   20483   20524   20578   20656   20713   20769
                    20818   20864   20906   20953   21128   21299   21321   21364   21385   21411   21465   21507   21546
                    21589   21630   21672   21709   21750   21784   21823   21868   21906   21945   21984   22026   22067
                    22105   22141   22183   22228   22488   22522   22551   22573   22595   22617   22639   22662   22688
                    22720   22961   23110   23362   23398   23427   23455   23485   23516   23547   23580   23604   23629
                    23870   23892   24012   24052   24072   24126   24147#
.$X     = 001030     64250#
```

```
ACCMAC  15048#  15647  15648  15649  15650  15651  15652  15653  15654  15655  15656  15657  15658
ADDTST  12168#  12847  12848  12849  12909  12910  12911  12912
APTSKP   6275#  10086  10930  10938  12746  24460  24487  24555  24573  24614  24731  24865  24900  24982
COMMEN   1577#
ENDCOM   1589#
ERROR    6312#  6577   6626   6627   6645   6646   6655   6662   6663   6689   6695   6701   6706   6727   6733
         6735   6745   6752   6762   6788   6793   6798   6803   6813   6824   6833   6844   6854   6865   6875
         6886   6897   6908   6918   6929   6949   6954   6959   6964   6998   7002   7004   7019   7029   7039
         7043   7056   7061   7075   7082   7100   7108   7127   7135   7154   7162   7177   7187   7201   7204
         7211   7213   7222   7225   7231   7233   7242   7249   7251   7273   7279   7301   7311   7334   7344
         7357   7359   7363   7365   7369   7384   7392   7419   7425   7440   7444   7461   7465   7475   7477
         7479   7492   7494   7497   7509   7511   7521   7526   7539   7544   7563   7581   7600   7618   7627
         7645   7648   7667   7669   7677   7680   7700   7703   7726   7733   7736   7754   7756   7776   7779
         7798   7800   7820   7822   7836   7848   7862   7864   7877   7883   7887   7895   7899   7916   7919
         7931   7933   7936   7941   7944   7959   7963   7978   7993   8009   8031   8034   8051   8066   8069
         8086   8088   8104   8107   8121   8134   8147   8157   8159   8169   8171   8180   8183   8185   8196
         8198   8203   8206   8208   8223   8225   8231   8234   8246   8248   8250   8262   8264   8267   8269
         8271   8283   8285   8287   8298   8300   8302   8313   8315   8317   8333   8335   8351   8354   8356
         8369   8371   8375   8377   8379   8393   8395   8397   8400   8403   8419   8421   8423   8439   8441
         8445   8447   8449   8467   8469   8471   8475   8477   8494   8496   8498   8502   8504   8520   8522
         8524   8527   8529   8550   8574   8597   8615   8631   8638   8657   8664   8673   8692   8701   8712
         8727   8733   8739   8758   8780   8796   8813   8827   8829   8843   8858   8860   8874   8889   8891
         8908   8910   8929   8948   8989   8991   8998   9003   9005   9011   9017   9024   9031   9062   9073
         9084   9094   9104   9112   9120   9131   9148   9150   9171   9178   9186   9193   9201   9207   9216
         9223   9243   9251   9258   9267   9273   9279   9287   9306   9315   9322   9330   9337   9355   9361
         9367   9371   9378   9387   9394   9401   9421   9429   9437   9446   9453   9461   9467   9476   9496
         9503   9510   9517   9527   9534   9541   9548   9568   9575   9581   9589   9598   9605   9612   9618
         9627   9634   9640   9649   9658   9666   9673   9683   9692   9697   9701   9703   9714   9716   9721
         9723   9733   9735   9740   9749   9754   9757   9764   9766   9786   9798   9819   9829   9843   9851
         9867   9874   9881   9885   9901   9909   9917   9919   9928   9930   9938   9940   9948   9950   9958
         9960   9968   9970   9984   9991   10001  10003  10005  10015  10017  10019  10029  10031  10033  10043
        10045  10047  10057  10059  10061  10071  10073  10075  10090  10106  10123  10126  10132  10150  10154
        10167  10170  10209  10225  10241  10267  10282  10305  10310  10318  10323  10333  10336  10340  10343
        10353  10357  10361  10364  10378  10384  10391  10397  10403  10414  10426  10436  10447  10490  10491
        10492  10493  10494  10495  10496  10497  10498  10499  10527  10532  10537  10543  10549  10555  10560
        10566  10571  10579  10587  10595  10603  10611  10614  10617  10620  10623  10625  10628  10631  10634
        10641  10649  10657  10662  10666  10693  10709  10719  10728  10730  10739  10751  10765  10769  10779
        10782  10795  10796  10808  10833  10837  10839  10856  10862  10864  10888  10904  10905  10921  10935
        10946  10975  10977  10979  10990  10992  11003  11014  11016  11033  11034  11036  11038  11040  11114
        11117  11119  11367  11370  11399  11402  11430  11433  11461  11464  11493  11496  11526  11529  11607
        11613  11614  11615  11616  11617  11618  11619  11620  11621  11622  11665  11668  11672  11675  11704
        11707  11711  11713  11743  11746  11750  11752  11865  11869  11870  11871  11872  11873  11874  11875
        11876  11882  11883  11884  11885  11886  11887  11888  11889  11890  11891  11988  11989  11990  11991
        11992  11993  11994  11995  11996  11997  11998  11999  12000  12001  12002  12003  12004  12005  12010
        12011  12012  12013  12014  12015  12016  12017  12018  12019  12030  12031  12032  12033  12034  12035
        12036  12037  12038  12039  12040  12041  12042  12043  12044  12045  12046  12051  12052  12053  12054
        12055  12056  12057  12058  12059  12060  12069  12071  12305  12324  12346  12356  12376  12453  12460
        12470  12478  12501  12518  12524  12531  12550  12559  12587  12598  12631  12642  12670  12680  12753
        12764  12774  12786  12847  12848  12849  12866  12886  12909  12910  12911  12912  12969  13001  13011
        13018  13039  13050  13098  13104  13111  13117  13158  13164  13171  13177  13255  13264  13270  13278
        13354  13363  13369  13377  13451  13488  13495  13517  13533  13541  13577  13607  13631  13640  13646
        13713  13784  13795  13805  13814  13824  13837  13848  13861  13895  13911  13926  13939  13940  13954
        13970  13985  14002  14046  14057  14067  14076  14086  14098  14109  14121  14144  14153  14177  14183
        14274  14287  14299  14312  15127  15151  15174  15183  15192  15200  15209  15217  15225  15233  15271
        15284  15288  15299  15323  15361  15363  15368  15373  15390  15392  15397  15402  15467  15493  15506
        15567  15593  15606  15694  15788  15855  15857  15879  15881  15885  15900  15942  15947  15951  16000
```

```
        16005   16009   16069   16077   16129   16135   16138   16187   16192   16248   16253   16306   16309   16355   16359
        16422   16439   16453   16467   16489   16493   16498   16555   16558   16574   16577   16591   16594   16610   16613
        16648   16652   16666   16670   16684   16688   16702   16706   16753   16756   16817   16820   16877   16895   16916
        16976   16979   16993   16996   17011   17014   17029   17032   17046   17049   17063   17066   17142   17146   17160
        17163   17179   17182   17197   17200   17214   17217   17231   17234   17301   17305   17319   17323   17337   17341
        17355   17358   17372   17375   17389   17393   17407   17411   17478   17482   17496   17500   17551   17554   17569
        17572   17638   17641   17660   17664   17681   17685   17699   17703   17717   17721   17806   17810   17821   17824
        17832   17836   17841   17857   17861   17874   17882   17886   17891   17907   17911   17922   17925   17933   17937
        17942   18002   18010   18014   18026   18036   18041   18052   18065   18082   18105   18110   18124   18129   18390
        18396   18435   18438   18454   18457   18471   18476   18483   18486   18659   18661   18663   18793   18797   18965
        18967   18969   19072   19076   19199   19203   19332   19336   19450   19453   19466   19470   19472   19590   19594
        19607   19610   19612   19816   19818   19821   19823   19825   20038   20045   20048   20078   20100   20121   20182
        20184   20187   20189   20191   20229   20251   20325   20332   20335   20361   20367   20371   20374   20379   20382
        20389   20392   20396   20401   20404   20408   20414   20417   20421   20427   20433   20441   20448   20455   20463
        20470   20506   20511   20520   20522   20556   20577   20610   20629   20646   20655   20697   20768   20816   20863
        20905   20952   21126   21297   21319   21352   21384   21409   21448   21506   21545   21588   21629   21671   21708
        21749   21783   21822   21867   21905   21944   21983   22025   22066   22104   22140   22182   22226   22483   22520
        22550   22572   22594   22616   22638   22661   22686   22710   22713   22956   23108   23357   23396   23425   23454
        23484   23515   23546   23579   23598   23623   23865   24001   24003   24006   24045   24064   24066   24095   24111
        24200   24209   24226   24231   24243   24251   24259   24270   24273   24276   24281   24293   24298   24314   24319
        24335   24342   24359   24392   24400   24403   24406   24415   24474   24480   24501   24541   24550   24563   24568
        24581   24589   24597   24607   24610   24613   24618   24626   24629   24632   24641   24650   24653   24656   24665
        24677   24682   24698   24703   24719   24726   24744   24758   24767   24780   24793   24797   24813   24818   24834
        24842   24859   24878   24890   24892   24910   24913   24925   24927   24938   24941   25003   25060   25078   25083
ESCAPE  17030#
GETPRI  13340#
GETSWR  17720#
MMAC1   150170#  15750   15751   15752   15753   15754
MMAC2   150250#  15775   15776   15777   15778   15779
MMAC3   150320#  15760   15761   15762   15763   15764
JNE      63450#   6561    6580    6582
MSG     150980#  151340#  151550#  152370#  153020#  153270#  154350#  155320#  156340#  157330#  158230#  159180#  159770#  160340#  161030#
        161590#  162190#  162810#  163280#  163840#  165330#  166280#  167280#  167920#  168560#  169520#  171180#  172800#  174570#  175270#
        176120#  177760#  179870#  181750#  184090#  184960#  186710#  188040#  189770#  190830#  192120#  193450#  194810#  196220#  198380#
        200590#  202090#  203510#  204870#  205280#  205820#  206580#  207150#  207710#  208200#  208660#  209080#  209550#  211300#  213000#
        213230#  213660#  213860#  222290#  226650#  226900#  227230#  229630#  231120#  235820#  236060#  236310#  238730#  238950#  240140#
MSG1    122920#  214130#  218240#  224910#
MSG10   124250#
MSG11   124310#
MSG12   124370#
MSG13   124860#
MSG14   125080#
MSG15   125330#
MSG16   125730#
MSG17   126140#
MSG2    123130#  233650#
MSG20   126530#
MSG21   126910#
MSG21A  127340#
MSG22   128150#
MSG23   128940#
MSG24   129140#
MSG25   130070#
MSG26   130130#
MSG27   130210#
MSG3    123320#
```

```
MSG30   130710
MSG31   131370
MSG32   132200
MSG33   133210
MSG34   134210
MSG35   134670
MSG36   135500
MSG36A  135900
MSG37   136120
MSG4    123610
MSG40   136550
MSG41   137330
MSG42   138690
MSG43   140100
MSG44   141280
MSG45   141600
MSG5    123940
MSG6    124130
MSG7    124190
MULT     45700
NEWTST   16350  62990  6551  6626  6627  6645  6646  6655  6662  6663  6685  6691  6697  6703  6722
          6729   6737  6747  6754  6784  6790  6795  6800  6805  6815  6825  6835  6846  6856  6867
          6877   6889  6899  6910  6920  6945  6951  6956  6961  6996  7016  7037  7053  7070  7094
          7120   7147  7169  7194  7214  7234  7267  7293  7327  7345  7379  7409  7435  7455  7467
          7480   7498  7512  7536  7555  7573  7591  7609  7636  7658  7689  7714  7745  7765  7788
          7810   7830  7842  7854  7874  7906  7926  7950  7971  7986  8002  8022  8043  8060  8080
          8096   8116  8129  8142  8149  8161  8173  8187  8211  8236  8252  8273  8289  8304  8325
          8343   8363  8386  8412  8433  8460  8487  8513  8543  8567  8590  8608  8624  8649  8682
          8721   8749  8770  8790  8806  8824  8838  8853  8870  8884  8903  8924  8943  8984  9050
          9141   9165  9179  9194  9208  9236  9260  9299  9308  9323  9349  9380  9413  9439  9469
          9489   9519  9560  9590  9619  9650  9686  9726  9743  9776  9808  9837  9859  9896  9911
          9921   9932  9942  9952  9962  9980  9993 10007 10021 10035 10049 10063 10085 10098 10117
         10142  10157 10181 10218 10234 10256 10299 10312 10325 10345 10374 10380 10386 10393 10399
         10409  10420 10428 10438 10523 10572 10612 10637 10644 10651 10678 10679 10681 10689 10690
         10691  10694 10696 10712 10721 10731 10742 10754 10771 10784 10800 10822 10878 10893 10909
         10929  10937 10952 10982 10997 11010 11026 11052 11063 12298 12317 12338 12369 12404 12416
         12422  12428 12434 12447 12495 12515 12544 12581 12621 12660 12702 12743 12832 12905 12937
         13010  13016 13030 13079 13145 13232 13333 13435 13481 13562 13596 13624 13670 13744 13881
         14021  14135 14168 14364 14405 14432 14522 14574 14664 14691 14729 14800 14881 15108 15139
         15165  15256 15309 15339 15440 15538 15645 15739 15830 15924 15984 16041 16111 16166 16226
         16287  16335 16398 16540 16634 16735 16800 16863 16962 17128 17287 17464 17536 17622 17790
         17993  18183 18417 18504 18678 18812 18985 19092 19221 19357 19489 19630 19846 20067 20217
         20357  20493 20534 20588 20663 20720 20776 20825 20871 20913 20960 21135 21305 21328 21371
         21395  21420 21467 21509 21547 21590 21631 21673 21710 21751 21785 21830 21869 21907 21946
         21986  22028 22068 22106 22142 22185 22234 22498 22525 22554 22574 22596 22618 22640 22672
         22697  22730 22969 23121 23372 23401 23431 23459 23489 23519 23551 23588 23612 23640 23880
         23901  24024 24055 24076 24196 24205 24213 24235 24247 24255 24265 24286 24305 24323 24346
         24388  24395 24419 24459 24486 24537 24546 24554 24572 24585 24593 24605 24621 24645 24670
         24689  24707 24730 24752 24774 24784 24804 24823 24847 24864 24882 24899 24918 24930 24981
         25013
NXTTST   63410  12569 12608 12649 12687 12999 13285 13384
POP      21460
PUSH     21380
REPORT   55160
RSET    150070  15131 15153 15234 15300 15324 15432 15530 15631 15730 15820 15916 15975 16032 16101
         16157  16217 16279 16326 16382 16625 16726 16791 16853 16951 17116 17279 17456 17526 17601
```

```
        17774   17985   18171   18405   18493   18668   18801   18974   19080   19208   19341   19477   19617   19834   20055
        20205   20349   20483   20524   20578   20656   20713   20769   20818   20864   20906   20953   21128   21299   21321
        21364   21385   21411   21465   21507   21546   21589   21630   21672   21709   21750   21784   21823   21868   21906
        21945   21984   22026   22067   22105   22141   22183   22228   22488   22522   22551   22573   22595   22617   22639
        22662   22688   22720   22961   23110   23362   23398   23427   23455   23485   23516   23547   23580   23604   23629
        23870   23892   24012   24052   24072   24126
SAVEMT   1174#   6421#   6484#
SETPRI   1303#
SETUP    1357#
SKIP     1736#
SLASH    1530#
STARS    1500#   6306#   6422    6424    6425    6426    6428    6458    6460    6464    6529    6551    6626    6627    6629
         6632    6645    6646    6648    6651    6655    6662    6663    6665    6685    6691    6697    6703    6708    6722
         6729    6737    6747    6754    6764    6784    6790    6795    6800    6805    6815    6825    6835    6846    6856
         6867    6877    6889    6899    6910    6920    6931    6945    6951    6956    6961    6965    6980    6984    6996
         7007    7016    7031    7037    7045    7053    7063    7070    7085    7094    7110    7120    7137    7147    7164
         7169    7189    7194    7214    7234    7253    7267    7281    7293    7313    7327    7345    7371    7379    7394
         7409    7427    7435    7446    7455    7467    7480    7498    7512    7528    7536    7547    7555    7565    7573
         7583    7591    7602    7609    7629    7636    7650    7658    7682    7689    7705    7714    7737    7745    7758
         7765    7781    7788    7803    7810    7824    7830    7838    7842    7850    7854    7866    7874    7901    7906
         7920    7926    7945    7950    7964    7971    7980    7986    7995    8002    8011    8022    8036    8043    8053
         8060    8071    8080    8089    8096    8108    8116    8122    8129    8135    8142    8149    8161    8173    8187
         8211    8236    8252    8273    8289    8304    8319    8325    8337    8343    8358    8363    8381    8386    8405
         8412    8425    8433    8451    8460    8479    8487    8506    8513    8531    8543    8558    8567    8580    8590
         8598    8608    8616    8624    8640    8649    8675    8682    8714    8721    8741    8749    8760    8770    8783
         8790    8798    8806    8817    8824    8831    8838    8846    8853    8863    8870    8877    8884    8894    8903
         8915    8924    8933    8943    8953    8984    9034    9050    9134    9141    9151    9165    9179    9194    9208
         9225    9236    9260    9290    9299    9308    9323    9339    9349    9380    9403    9413    9439    9469    9479
         9489    9519    9550    9560    9590    9619    9650    9686    9726    9743    9768    9776    9799    9808    9830
         9837    9852    9859    9887    9896    9911    9921    9932    9942    9952    9962    9972    9980    9993   10007
        10021   10035   10049   10063   10077   10085   10093   10098   10110   10117   10137   10142   10157   10174   10181
        10212   10218   10228   10234   10242   10256   10289   10299   10312   10325   10345   10367   10374   10380   10386
        10393   10399   10404   10408   10409   10418   10420   10428   10438   10484   10488   10500#  10523   10572   10612
        10637   10644   10651   10678   10679   10681   10689   10690   10691   10694   10696   10712   10721   10731   10742
        10754   10771   10784   10800   10822   10878   10893   10909   10929   10937   10952   10982   10997   11010   11026
        11052   11063   12298   12317   12338   12369   12381   12392   12404   12416   12422   12428   12434   12447   12495
        12515   12544   12581   12621   12660   12702   12743   12832   12905   12937   13010   13016   13030   13059   13069
        13079   13145   13202   13217   13232   13333   13435   13481   13562   13596   13624   13670   13725   13730   13744
        13881   14021   14135   14168   14194   14201   14213   14219   14229   14237   14258   14266   14321   14336   14338
        14364   14405   14432   14522   14574   14664   14691   14729   14800   14881   15108   15139   15165   15256   15309
        15339   15440   15538   15645   15739   15830   15924   15984   16041   16111   16166   16226   16287   16335   16398
        16540   16634   16735   16800   16863   16962   17128   17287   17464   17536   17622   17790   17993   18183   18417
        18504   18678   18812   18985   19092   19221   19357   19489   19630   19846   20067   20217   20357   20493   20534
        20588   20663   20720   20776   20825   20871   20913   20960   21135   21305   21328   21371   21395   21420   21467
        21509   21547   21590   21631   21673   21710   21751   21785   21830   21869   21907   21946   21986   22028   22068
        22106   22142   22185   22234   22498   22525   22554   22574   22596   22618   22640   22672   22697   22730   22969
        23121   23372   23401   23431   23459   23489   23519   23551   23588   23612   23640   23880   23901   24024   24055
        24076   24143   24196   24205   24213   24235   24247   24255   24265   24286   24305   24323   24346   24388   24395
        24419   24459   24486   24537   24546   24554   24572   24585   24593   24605   24621   24645   24670   24689   24707
        24730   24752   24774   24784   24804   24823   24847   24864   24882   24899   24918   24930   24981   25013   25142
        25145   25173   25176
SWRSU    1470#
TIMMSG  12128#
TIMTST  12144#  12417   12423   12429   12435
TRAPTS   6587#   6626    6627    6645    6646    6655    6662    6663
TYPBIN   2083#
```

```
TYPDEC   20530#
TYPNAM   18250#
TYPNUM   20200#
TYPOCS   19730#
TYPOCT   19360#
TYPTXT   18910#
USER     12101#  12272
VTRP     10668#  10678   10679   10689   10690   10691   10694
WMSG     12203#
WTST     12227#  13011   13018
$ABORT    2861#   6421#  25172
$ASHC    11237#  11869   11870   11871   11872   11873   11874   11875   11876
$ASHC$   11280#  11882   11883   11884   11885   11886   11887   11888   11889   11890   11891
$ASH$    11211#  11613   11614   11615   11616   11617   11618   11619   11620   11621   11622
$DIV     11934#  12030   12031   12032   12033   12034   12035   12036   12037   12038   12039   12040   12041   12042   12043
         12044   12045   12046   12051   12052   12053   12054   12055   12056   12057   12058   12059   12060
$MUL     11899#  11988   11989   11990   11991   11992   11993   11994   11995   11996   11997   11998   11999   12000   12001
         12002   12003   12004   12005   12010   12011   12012   12013   12014   12015   12016   12017   12018   12019
$SKIP     6283#  10086   10930   10938   12746   24460   24487   24555   24373   24614   24731   24865   24900   24982
$$ERCD    6333#   6577    6626    6627    6645    6646    6655    6662    6663    6689    6695    6701    6706    6727    6733
          6735    6745    6752    6762    6788    6793    6798    6803    6813    6824    6833    6844    6854    6865    6875
          6886    6897    6908    6918    6929    6949    6954    6959    6964    6998    7002    7004    7019    7029    7039
          7043    7056    7061    7075    7082    7100    7108    7127    7135    7154    7162    7177    7187    7201    7204
          7211    7213    7222    7225    7231    7233    7242    7249    7251    7273    7279    7301    7311    7334    7344
          7357    7359    7363    7365    7369    7384    7392    7419    7425    7440    7444    7461    7465    7475    7477
          7479    7492    7494    7497    7509    7511    7521    7526    7539    7544    7563    7581    7600    7618    7627
          7645    7648    7667    7669    7677    7680    7700    7703    7726    7733    7736    7754    7756    7776    7779
          7798    7800    7820    7822    7836    7848    7862    7864    7877    7883    7887    7895    7899    7916    7919
          7931    7933    7936    7941    7944    7959    7963    7978    7993    8009    8031    8034    8051    8066    8069
          8086    8088    8104    8107    8121    8134    8147    8157    8159    8169    8171    8180    8183    8185    8196
          8198    8203    8206    8208    8223    8225    8231    8234    8246    8248    8250    8262    8264    8267    8269
          8271    8283    8285    8287    8298    8300    8302    8313    8315    8317    8333    8335    8351    8354    8356
          8369    8371    8375    8377    8379    8393    8395    8397    8400    8403    8419    8421    8423    8439    8441
          8445    8447    8449    8467    8469    8471    8475    8477    8494    8496    8498    8502    8504    8520    8522
          8524    8527    8529    8550    8574    8597    8615    8631    8638    8657    8664    8673    8692    8701    8712
          8727    8733    8739    8758    8780    8796    8813    8827    8829    8843    8858    8860    8874    8889    8891
          8908    8910    8929    8948    8989    8991    8998    9003    9005    9011    9017    9024    9031    9062    9073
          9084    9094    9104    9112    9120    9131    9148    9150    9171    9178    9186    9193    9201    9207    9216
          9223    9243    9251    9258    9267    9273    9279    9287    9306    9315    9322    9330    9337    9355    9361
          9367    9371    9378    9387    9394    9401    9421    9429    9437    9446    9453    9461    9467    9476    9496
          9503    9510    9517    9527    9534    9541    9548    9568    9575    9581    9589    9598    9605    9612    9618
          9627    9634    9640    9649    9658    9666    9673    9683    9692    9697    9701    9703    9714    9716    9721
          9723    9733    9735    9740    9749    9754    9757    9764    9766    9786    9798    9819    9829    9843    9851
          9867    9874    9881    9885    9901    9909    9917    9919    9928    9930    9938    9940    9948    9950    9958
          9960    9968    9970    9984    9991   10001   10003   10005   10015   10017   10019   10029   10031   10033   10043
         10045   10047   10057   10059   10061   10071   10073   10075   10090   10106   10123   10126   10132   10150   10154
         10167   10170   10209   10225   10241   10267   10282   10305   10310   10318   10323   10333   10336   10340   10343
         10353   10357   10361   10364   10378   10384   10391   10397   10403   10414   10426   10436   10447   10490   10491
         10492   10493   10494   10495   10496   10497   10498   10499   10527   10532   10537   10543   10549   10555   10560
         10566   10571   10579   10587   10595   10603   10611   10614   10617   10620   10623   10625   10628   10631   10634
         10641   10649   10657   10662   10666   10693   10709   10719   10728   10730   10739   10751   10765   10769   10779
         10782   10795   10796   10808   10833   10837   10839   10856   10862   10864   10888   10904   10905   10921   10935
         10946   10975   10977   10979   10990   10992   11003   11014   11016   11033   11034   11036   11038   11040   11114
         11117   11119   11367   11370   11399   11402   11430   11433   11461   11464   11493   11496   11526   11529   11607
         11613   11614   11615   11616   11617   11618   11619   11620   11621   11622   11665   11668   11672   11675   11704
         11707   11711   11713   11743   11746   11750   11752   11865   11869   11870   11871   11872   11873   11874   11875
```

```
          11876  11882  11883  11884  11885  11886  11887  11888  11889  11890  11891  11988  11989  11990  11991
          11992  11993  11994  11995  11996  11997  11998  11999  12000  12001  12002  12003  12004  12005  12010
          12011  12012  12013  12014  12015  12016  12017  12018  12019  12030  12031  12032  12033  12034  12035
          12036  12037  12038  12039  12040  12041  12042  12043  12044  12045  12046  12051  12052  12053  12054
          12055  12056  12057  12058  12059  12060  12069  12071  12305  12324  12346  12356  12376  12453  12460
          12470  12478  12501  12518  12524  12531  12550  12559  12587  12598  12631  12642  12670  12680  12753
          12764  12774  12786  12847  12848  12849  12866  12886  12909  12910  12911  12912  12969  13001  13011
          13018  13039  13050  13098  13104  13111  13117  13158  13164  13171  13177  13255  13264  13270  13278
          13354  13363  13369  13377  13451  13488  13495  13517  13533  13541  13577  13607  13631  13640  13646
          13713  13784  13795  13805  13814  13824  13837  13848  13861  13895  13911  13926  13939  13940  13954
          13970  13985  14002  14046  14057  14067  14076  14086  14098  14109  14121  14144  14153  14177  14183
          14274  14287  14299  14312  15127  15151  15174  15183  15192  15200  15209  15217  15225  15233  15271
          15284  15288  15299  15323  15361  15363  15368  15373  15390  15392  15397  15402  15467  15493  15506
          15567  15593  15606  15694  15788  15855  15857  15879  15881  15885  15900  15942  15947  15951  16000
          16005  16009  16069  16077  16129  16135  16138  16187  16192  16248  16253  16306  16309  16355  16359
          16422  16439  16453  16467  16489  16493  16498  16555  16558  16574  16577  16591  16594  16610  16613
          16648  16652  16666  16670  16684  16688  16702  16706  16753  16756  16817  16820  16877  16895  16916
          16976  16979  16993  16996  17011  17014  17029  17032  17046  17049  17063  17066  17142  17146  17160
          17163  17179  17182  17197  17200  17214  17217  17231  17234  17301  17305  17319  17323  17337  17341
          17355  17358  17372  17375  17389  17393  17407  17411  17478  17482  17496  17500  17551  17554  17569
          17572  17638  17641  17660  17664  17681  17685  17699  17703  17717  17721  17806  17810  17821  17824
          17832  17836  17841  17857  17861  17874  17882  17886  17891  17907  17911  17922  17925  17933  17937
          17942  18002  18010  18014  18026  18036  18041  18052  18065  18082  18105  18110  18124  18129  18390
          18396  18435  18438  18454  18457  18471  18476  18483  18486  18659  18661  18663  18793  18797  18965
          18967  18969  19072  19076  19199  19203  19332  19336  19450  19453  19466  19470  19472  19590  19594
          19607  19610  19612  19816  19818  19821  19823  19825  20038  20045  20048  20078  20100  20121  20182
          20184  20187  20189  20191  20229  20251  20325  20332  20335  20361  20367  20371  20374  20379  20382
          20389  20392  20396  20401  20404  20408  20414  20417  20421  20427  20433  20441  20448  20455  20463
          20470  20506  20511  20520  20522  20556  20577  20610  20629  20646  20655  20697  20768  20816  20863
          20905  20952  21126  21297  21319  21352  21384  21409  21448  21506  21545  21588  21629  21671  21708
          21749  21783  21822  21867  21905  21944  21983  22025  22066  22104  22140  22182  22226  22483  22520
          22550  22572  22594  22616  22638  22661  22686  22710  22713  22956  23108  23357  23396  23425  23454
          23484  23515  23546  23579  23598  23623  23865  24001  24003  24006  24045  24064  24066  24095  24111
          24200  24209  24226  24231  24243  24251  24259  24270  24273  24276  24281  24293  24298  24314  24319
          24335  24342  24359  24392  24400  24403  24406  24415  24474  24480  24501  24541  24550  24563  24568
          24581  24589  24597  24607  24610  24613  24618  24626  24629  24632  24641  24650  24653  24656  24665
          24677  24682  24698  24703  24719  24726  24744  24758  24767  24780  24793  24797  24813  24818  24834
          24842  24859  24878  24890  24892  24910  24913  24925  24927  24938  24941  25003  25060  25078  25083
$$ERRO    63280   6689   6695   6701   6706   6727   6735   6745   6752   6762   6788   6793   6798   6803   6813
           6824   6833   6844   6854   6865   6875   6886   6897   6908   6918   6929   6949   6954   6959   6964
           7004   7029   7043   7061   7082   7108   7135   7162   7187   7213   7233   7251   7279   7311   7344
           7369   7392   7425   7444   7465   7479   7497   7511   7526   7544   7563   7581   7600   7627   7648
           7680   7703   7736   7756   7779   7800   7822   7836   7848   7864   7899   7919   7944   7963   7978
           7993   8009   8034   8051   8069   8088   8107   8121   8134   8147   8159   8171   8185   8208   8234
           8250   8271   8287   8302   8317   8335   8356   8379   8403   8423   8449   8477   8504   8529   8550
           8574   8597   8615   8638   8673   8712   8739   8758   8780   8796   8813   8829   8843   8860   8874
           8891   8910   8929   8948   9031   9131   9150   9178   9193   9207   9223   9258   9287   9306   9322
           9337   9378   9401   9437   9467   9476   9517   9548   9589   9618   9649   9683   9723   9740   9766
           9798   9829   9851   9885   9909   9919   9930   9940   9950   9960   9970   9991  10005  10019  10033
          10047  10061  10075  10090  10170  10241  10310  10323  10343  10364  10378  10384  10391  10397  10403
          10414  10426  10436  10571  10611  10634  10649  10693  10730  10739  10751  10782  10935  10992  11003
          11016  12356  12376  12531  12886  13098  13255  13354  13607  13939  20511  24243  24581  24618  24780
$$ESCA    17160
$$NEWT    16710  62860   6551   6626   6627   6645   6646   6655   6662   6663   6685   6691   6697   6703   6722
           6729   6737   6747   6754   6784   6790   6795   6800   6805   6815   6825   6835   6846   6856   6867
           6877   6889   6899   6910   6920   6945   6951   6956   6961   6996   7016   7037   7053   7070   7094
```

```
        7120    7147    7169    7194    7214    7234    7267    7293    7327    7345    7379    7409    7435    7455    7467
        7480    7498    7512    7536    7555    7573    7591    7609    7636    7658    7689    7714    7745    7765    7788
        7810    7830    7842    7854    7874    7906    7926    7950    7971    7986    8002    8022    8043    8060    8080
        8096    8116    8129    8142    8149    8161    8173    8187    8211    8236    8252    8273    8289    8304    8325
        8343    8363    8386    8412    8433    8460    8487    8513    8543    8567    8590    8608    8624    8649    8682
        8721    8749    8770    8790    8806    8824    8838    8853    8870    8884    8903    8924    8943    8984    9050
        9141    9165    9179    9194    9208    9236    9260    9299    9308    9323    9349    9380    9413    9439    9469
        9489    9519    9560    9590    9619    9650    9686    9726    9743    9776    9808    9837    9859    9896    9911
        9921    9932    9942    9952    9962    9980    9993   10007   10021   10035   10049   10063   10085   10098   10117
       10142   10157   10181   10218   10234   10256   10299   10312   10325   10345   10374   10380   10386   10393   10399
       10409   10420   10428   10438   10523   10572   10612   10637   10644   10651   10678   10679   10681   10689   10690
       10691   10694   10696   10712   10721   10731   10742   10754   10771   10784   10800   10822   10878   10893   10909
       10929   10937   10952   10982   10997   11010   11026   11052   11063   12298   12317   12338   12369   12404   12416
       12422   12428   12434   12447   12495   12515   12544   12581   12621   12660   12702   12743   12832   12905   12937
       13010   13016   13030   13079   13145   13232   13333   13435   13481   13562   13596   13624   13670   13744   13881
       14021   14135   14168   14364   14405   14432   14522   14574   14664   14691   14729   14800   14881   15108   15139
       15165   15256   15309   15339   15440   15538   15645   15739   15830   15924   15984   16041   16111   16166   16226
       16287   16335   16398   16540   16634   16735   16800   16863   16962   17128   17287   17464   17536   17622   17790
       17993   18183   18417   18504   18678   18812   18985   19092   19221   19357   19489   19630   19846   20067   20217
       20357   20493   20534   20588   20663   20720   20776   20825   20871   20913   20960   21135   21305   21328   21371
       21395   21420   21467   21509   21547   21590   21631   21673   21710   21751   21785   21830   21869   21907   21946
       21986   22028   22068   22106   22142   22185   22234   22498   22525   22554   22574   22596   22618   22640   22672
       22697   22730   22969   23121   23372   23401   23431   23459   23489   23519   23551   23588   23612   23640   23880
       23901   24024   24055   24076   24196   24205   24213   24235   24247   24255   24265   24286   24305   24323   24346
       24388   24395   24419   24459   24486   24537   24546   24554   24572   24585   24593   24605   24621   24645   24670
       24689   24707   24730   24752   24774   24784   24804   24823   24847   24864   24882   24899   24918   24930   24981
       25013
$$SKIP  1749#
.EQUAT   198#
.HEADE    74#
.KT11    340#  12078#  12081
.SETUP  1232#   6421#   6484
.SWRHI   115#
.$ACT1  5129#   6421#   6422
.$APTB  5172#   6421#   6424
.$APTH  5427#   6421#   6425
.$APTY  5599#
.$ASTA  5472#
.$CATC   924#
.$CMTA  1033#
.$DB2D  4765#
.$DB2O  4886#
.$DIV   4669#
.$EOP   2204#
.$ERRO  2683#
.$ERRT  2958#
.$MULT  4607#
.$POWE  4283#
.$RAND  4357#
.$RDDE  3957#
.$RDOC  3867#
.$READ  3466#
.$R2AZ  5027#
.$SAVE  4031#
.$SB2D  4848#
.$SB2O  4947#
```

```
.$SCOP   24380
.$SIZE   44090
.$SUPR   49840
.$TRAP   41320
.$TYPB   33600
.$TYPD   32840
.$TYPE   30440
.$TYPO   31890
.$40CA    9620
.1170     5180


. ABS.  133657     000


 ERRORS DETECTED:  0

 CJKL58,CJKL58/CRF/NL:TOC=ORION.SML,CJKL58.P11
 RUN-TIME: 72 86 12 SECONDS
 RUN-TIME RATIO: 488/171=2.8
 CORE USED:  51K  (102 PAGES)
```